# Programming without coding technology

# تكنولوجيا البرمجة بدون كود

(1) Mahmoud Programming Language

(2) RPWI Environment

(3) DoubleS (Super Server) Paradigm

(١) لغة البرمجة محمود

(٢) بيئة البرمجة بدون كود

(٣) نمط البرمجة الخادم الممتاز




Version 1.0
(Stable) Rev. 8

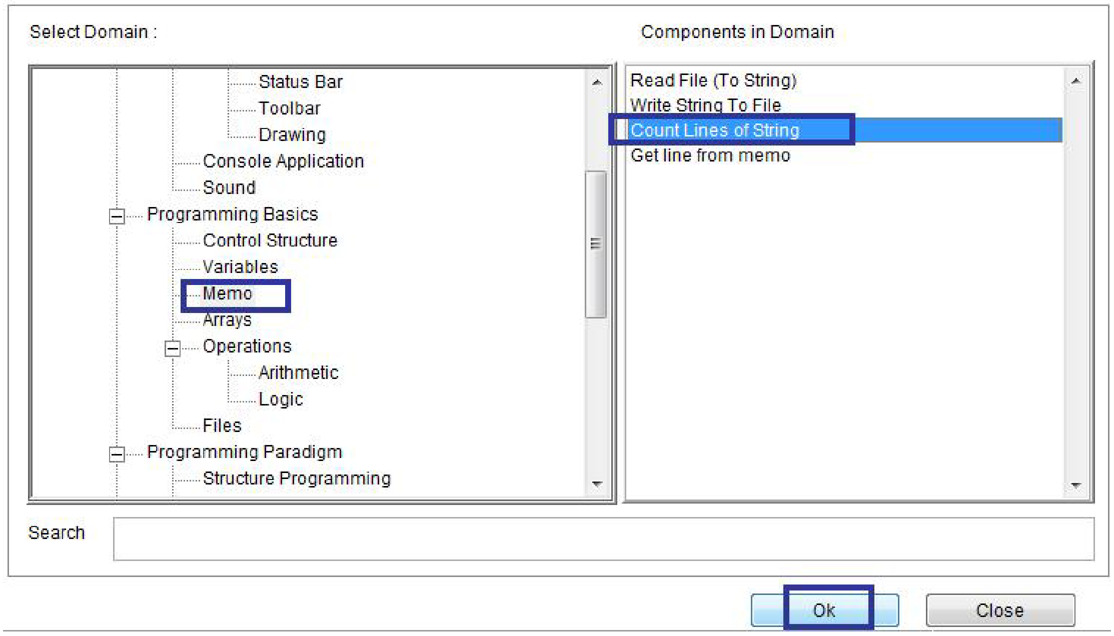
الاصدارة الاولى
المراجعة الثامنة

http://www.sourceforge.net/projects/doublesvsoop

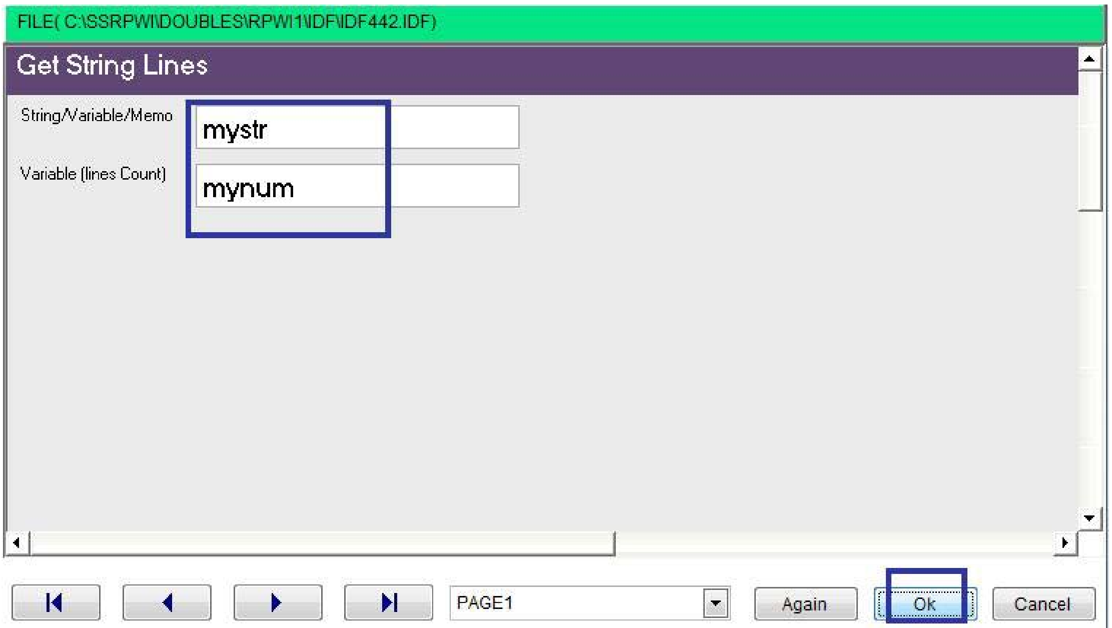
By
Mahmoud Fayed
msfclipper@users.sourceforge.net

# جدول المحتويات

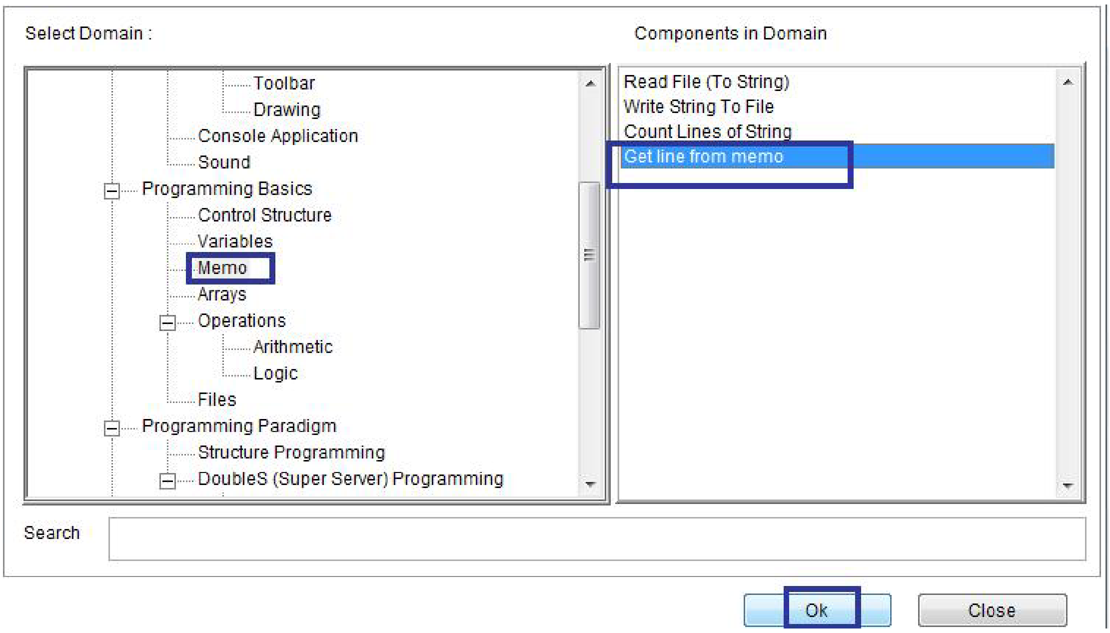
`



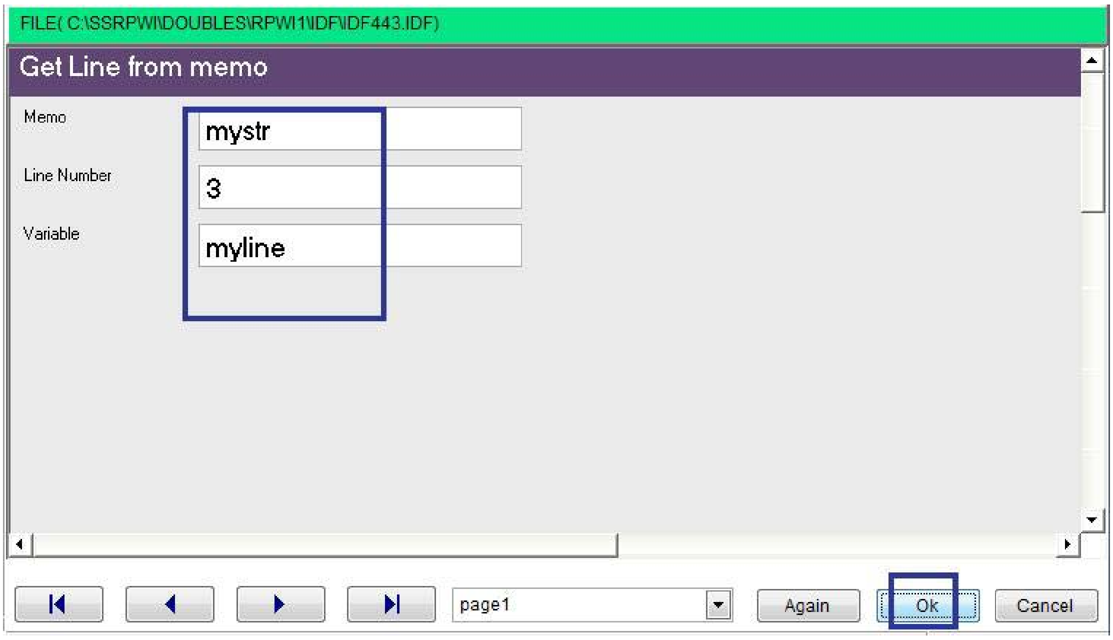

Domain (Memo) – Variable (Count Lines of String)



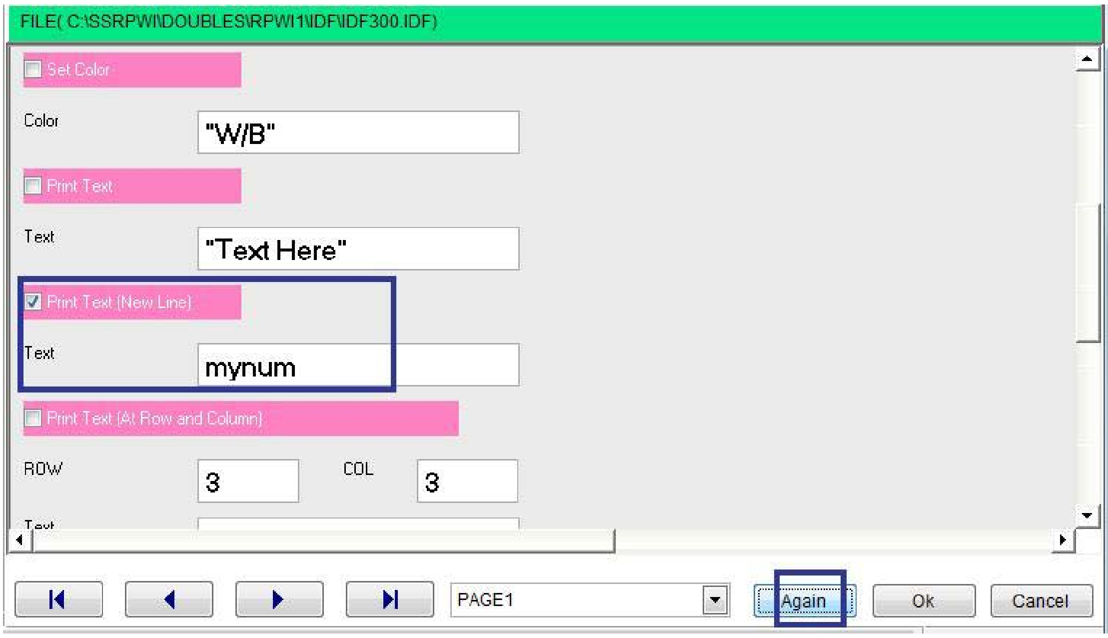

Interaction Page



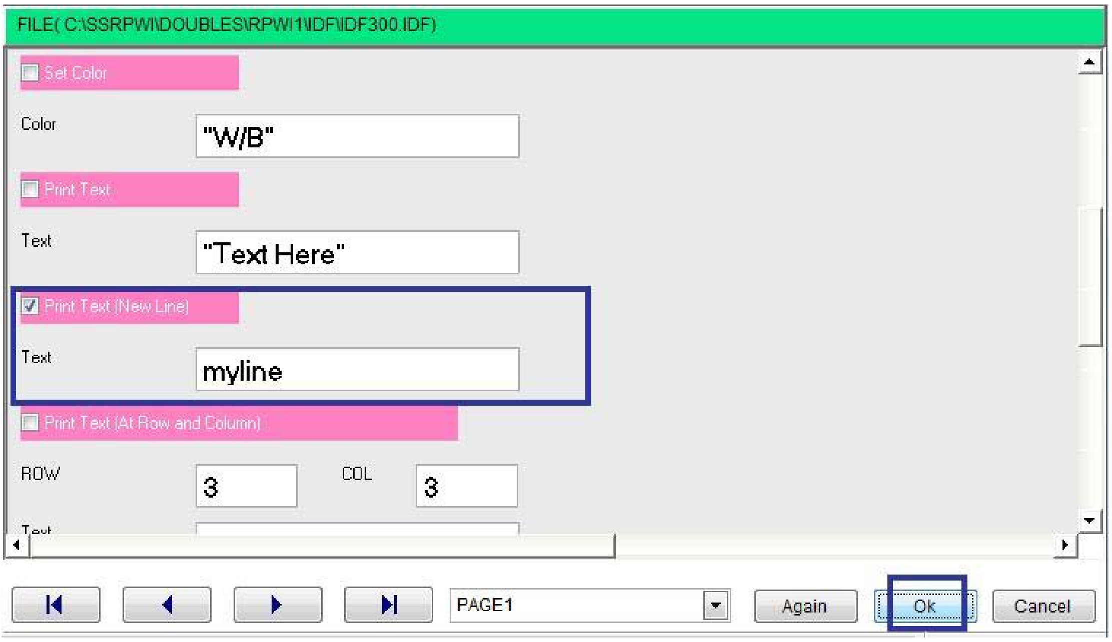

Domain (Memo) – Component (Get line from memo)



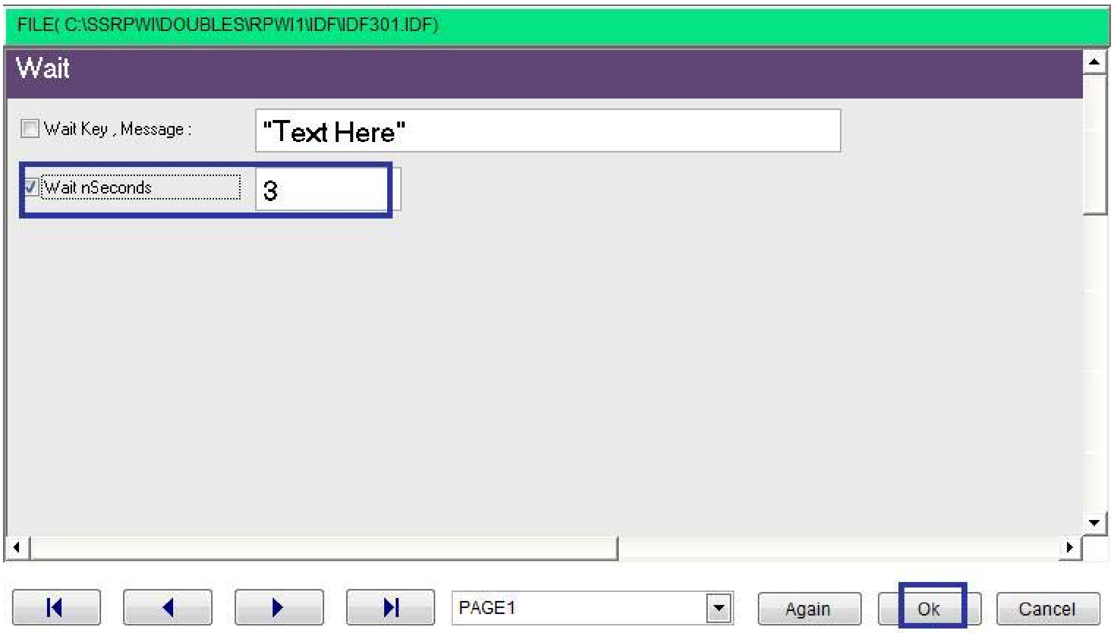

Interaction Page



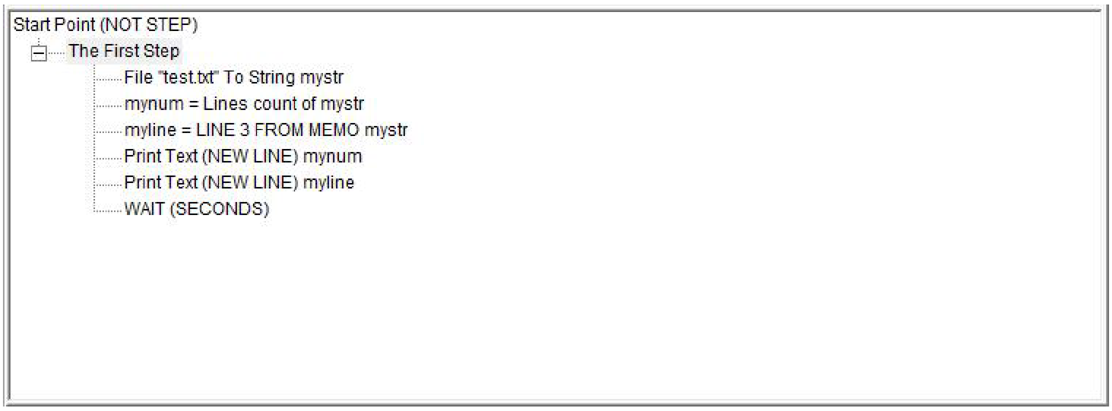
Interaction Page



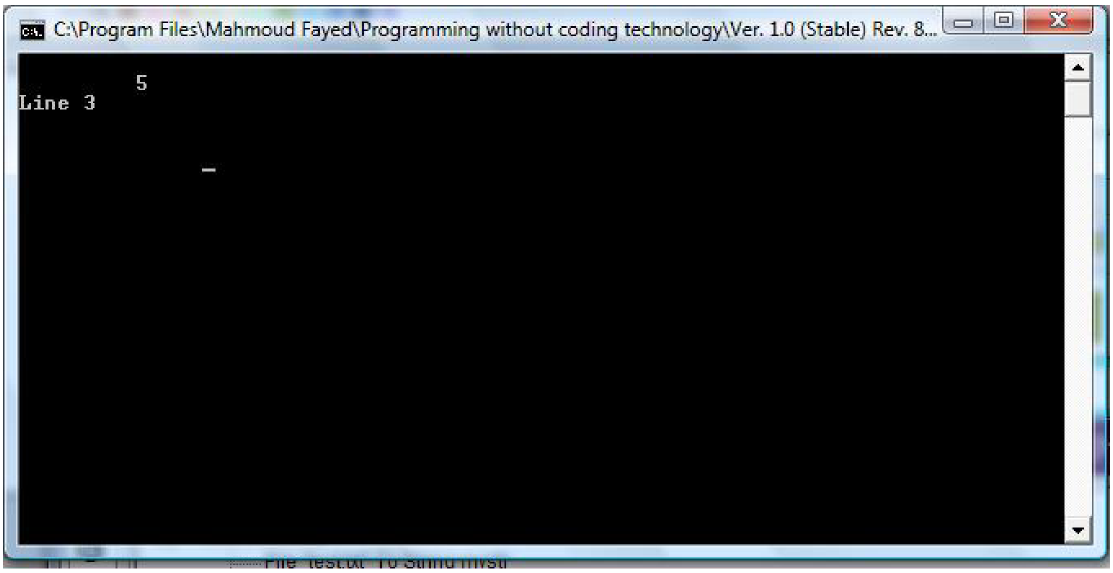
Interaction Page




Interaction Page




Final Steps Tree



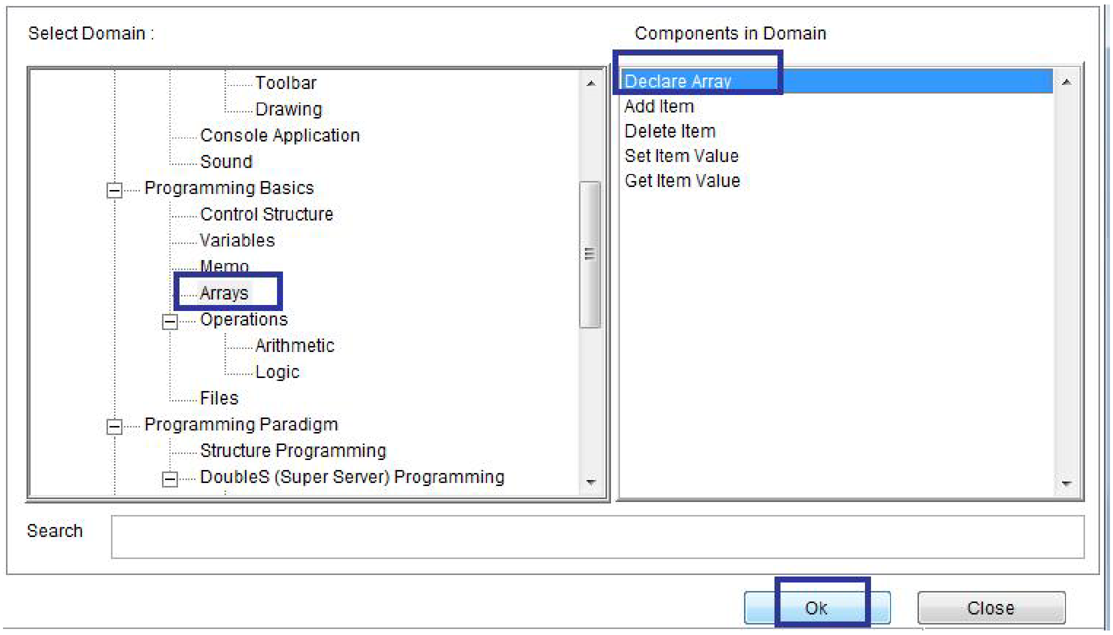

Final Application

# Arrays

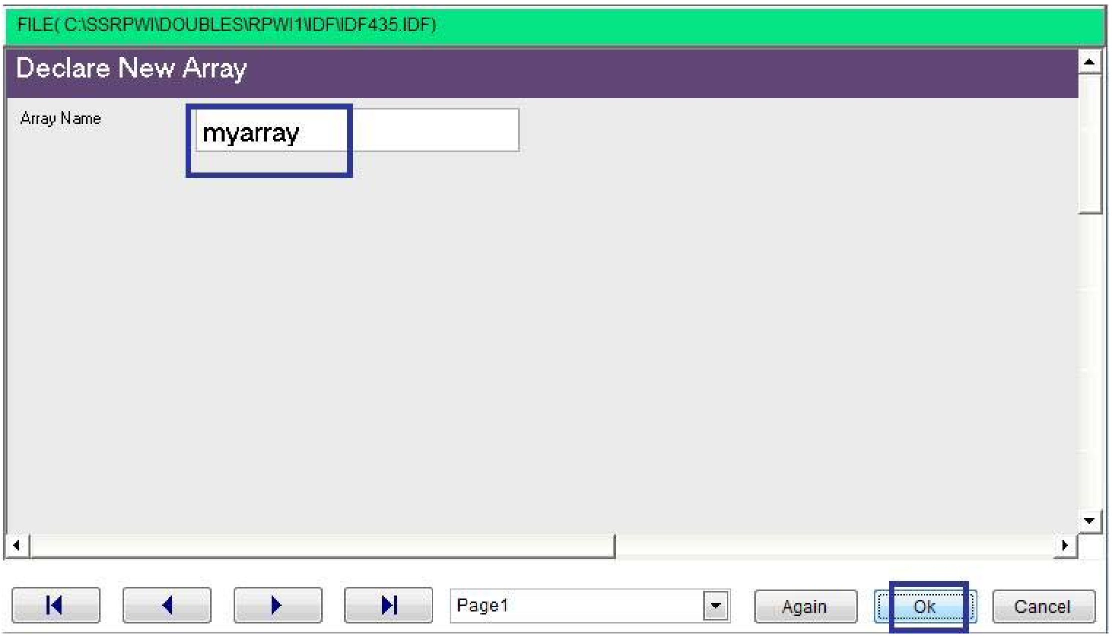
Components

- Declare array
- Add item
- Delete item
- Set item value
- Get item value

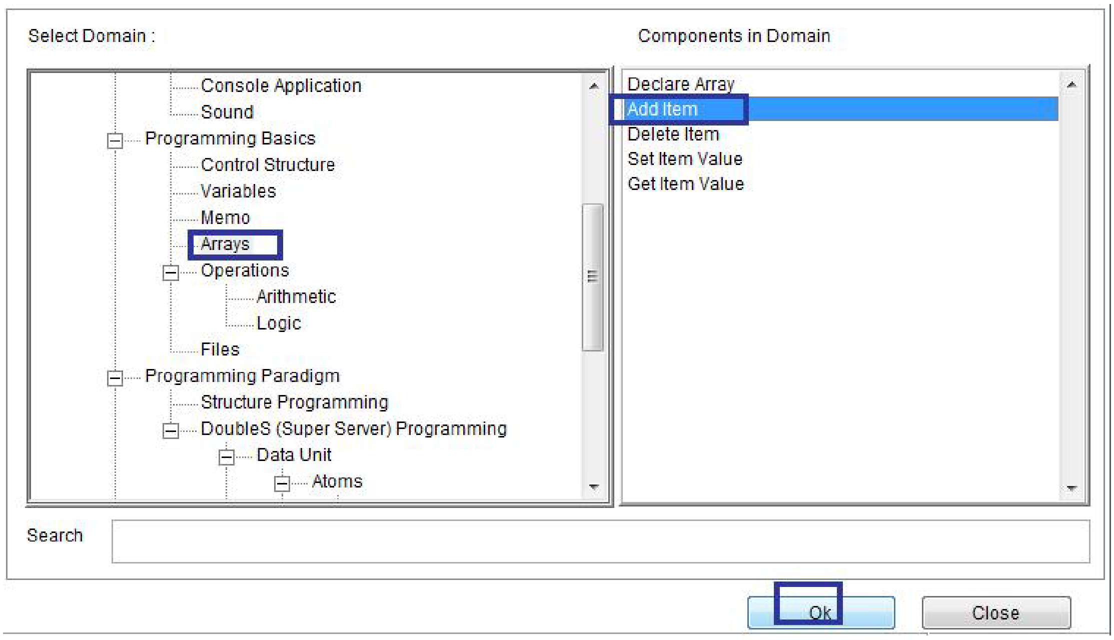
Example - Screen shots:-



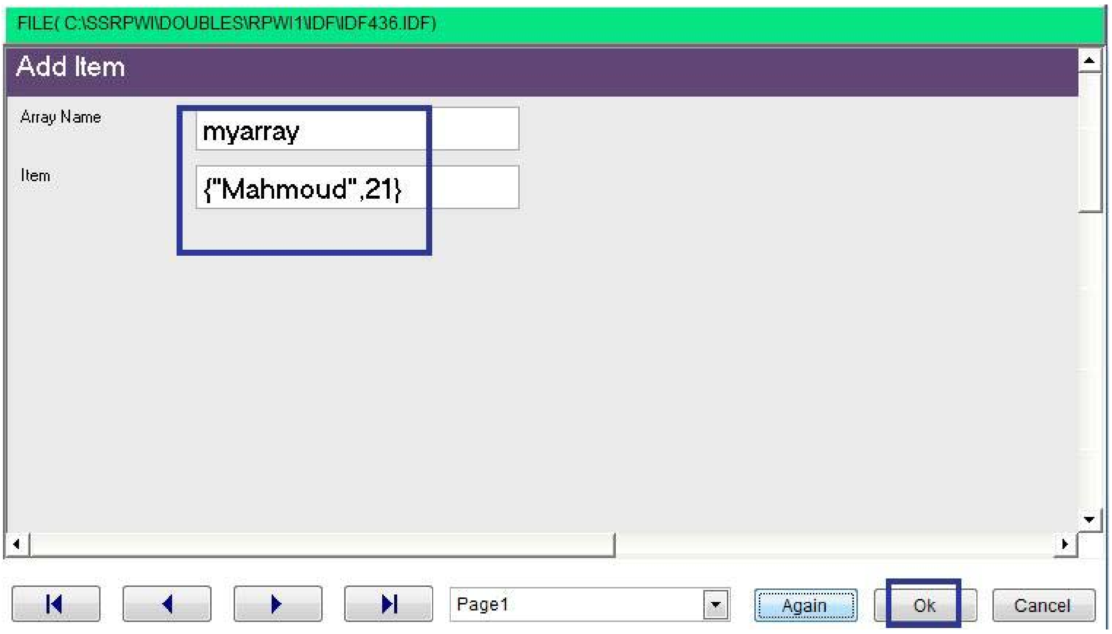

Domain (Arrays) Component (Declare array)



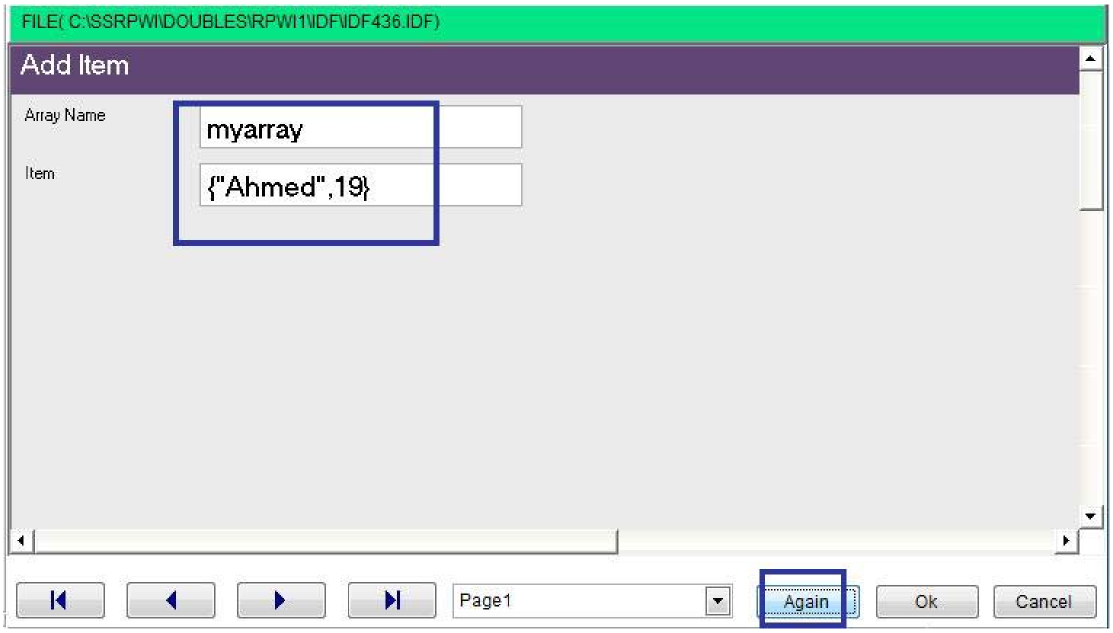

Interaction Page



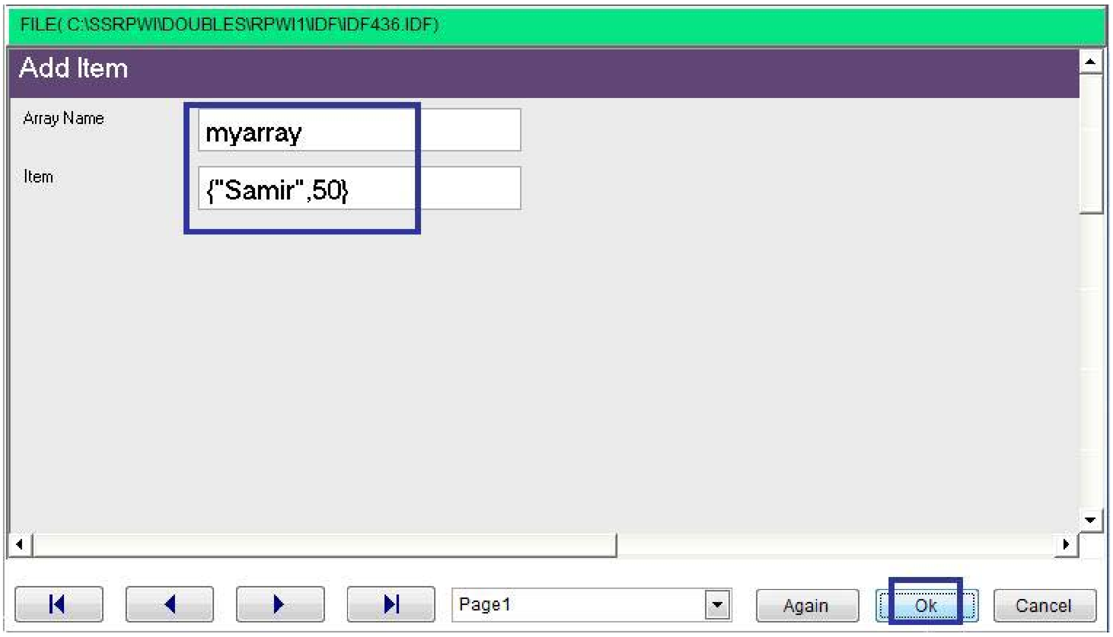

Domain (Arrays) Component (Add Item)



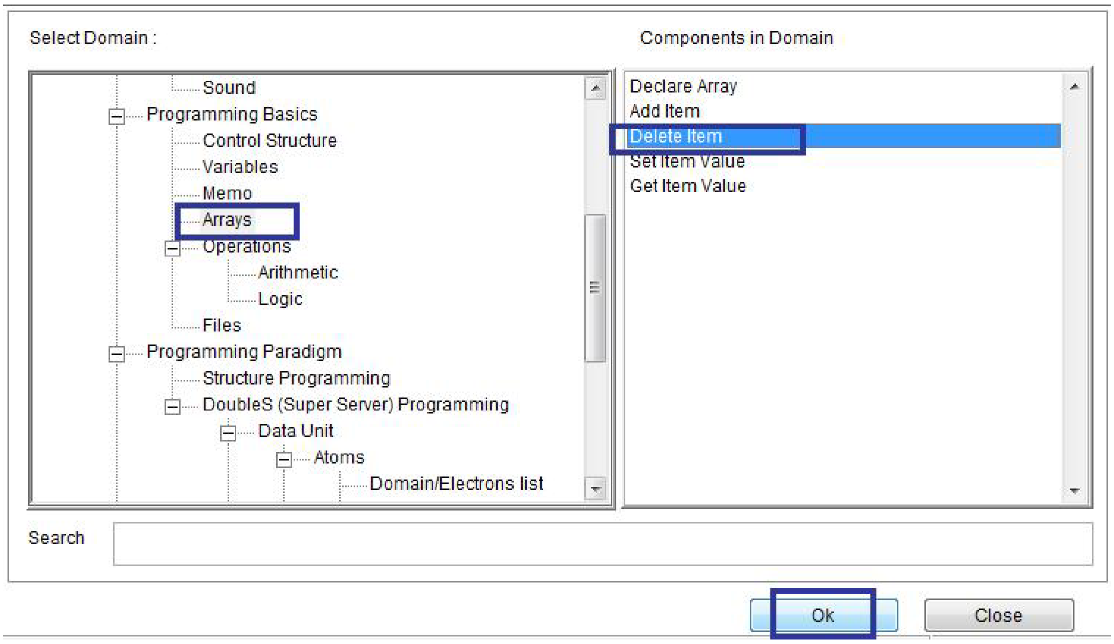

Interaction Page



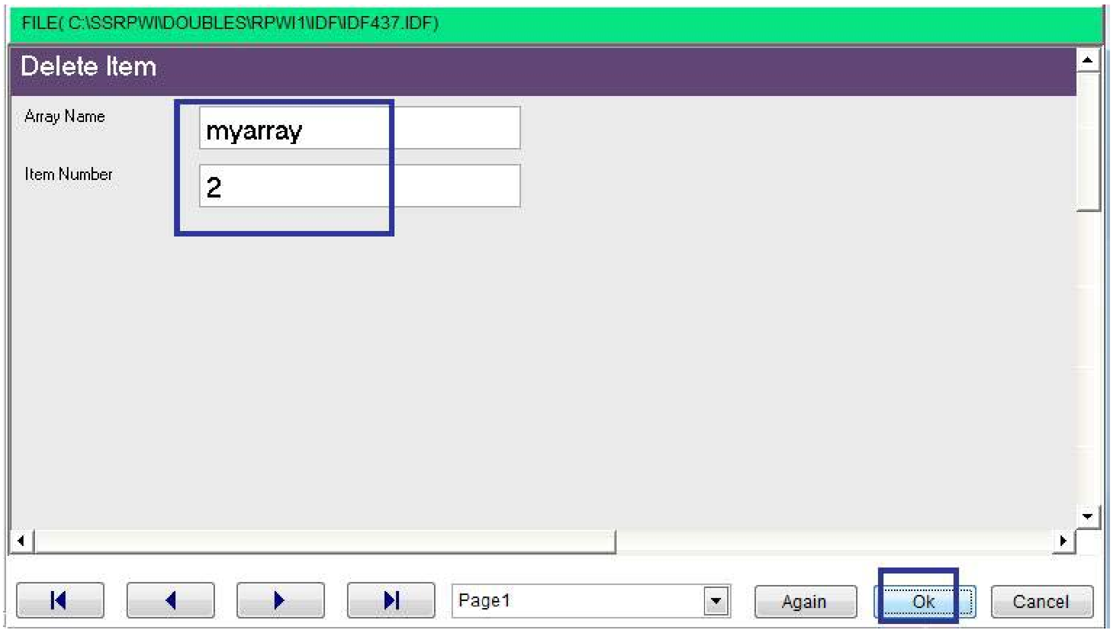
Interaction Page



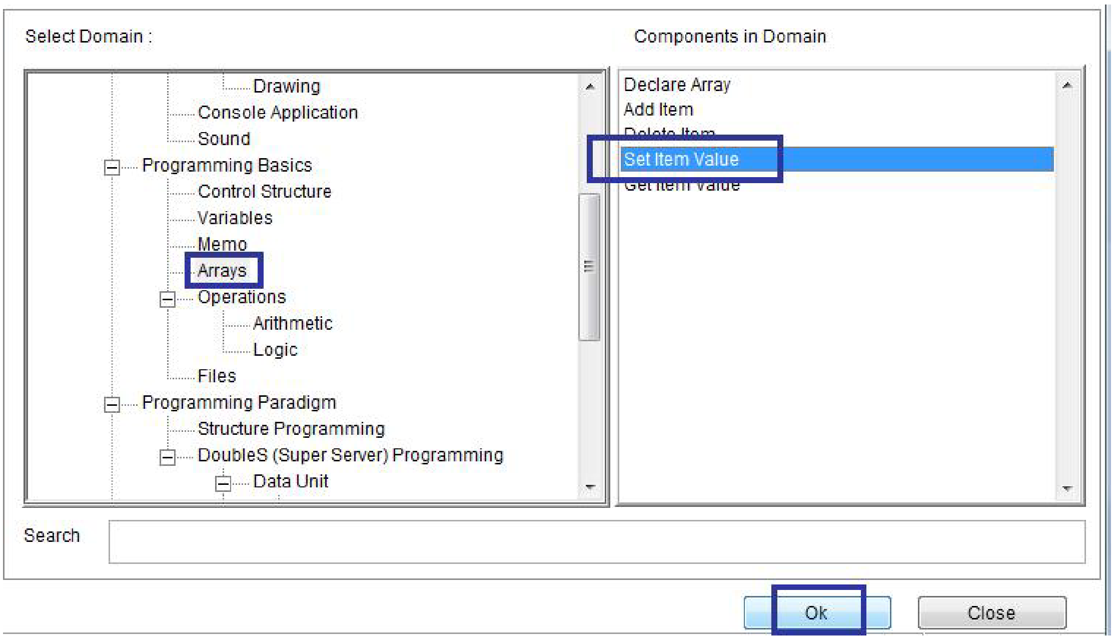
Interaction Page



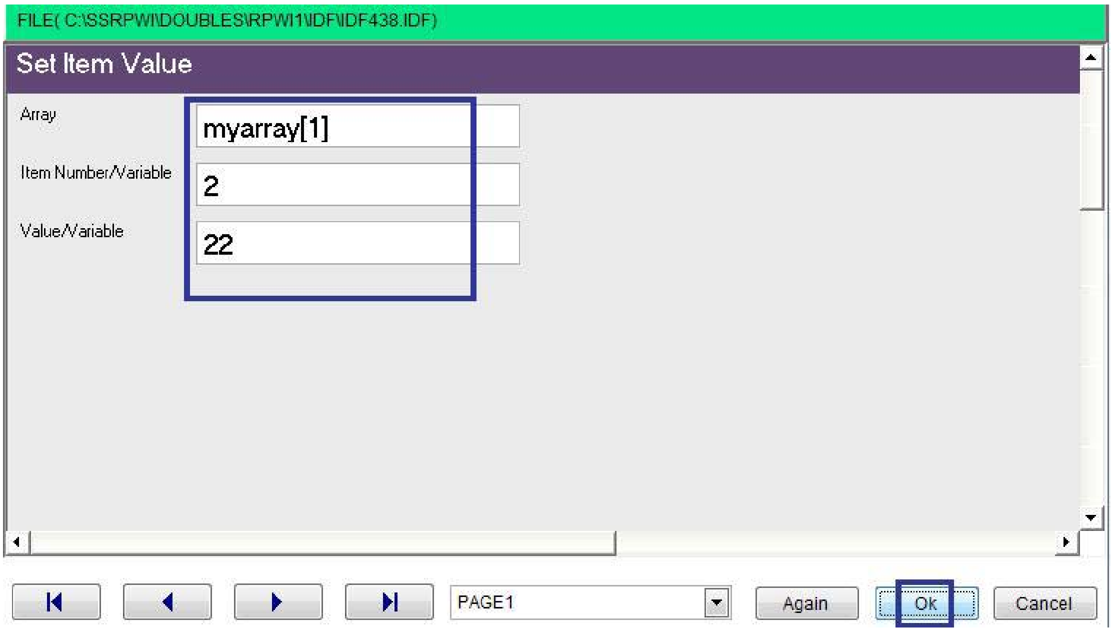

Domain (Arrays) Component (Delete Item)



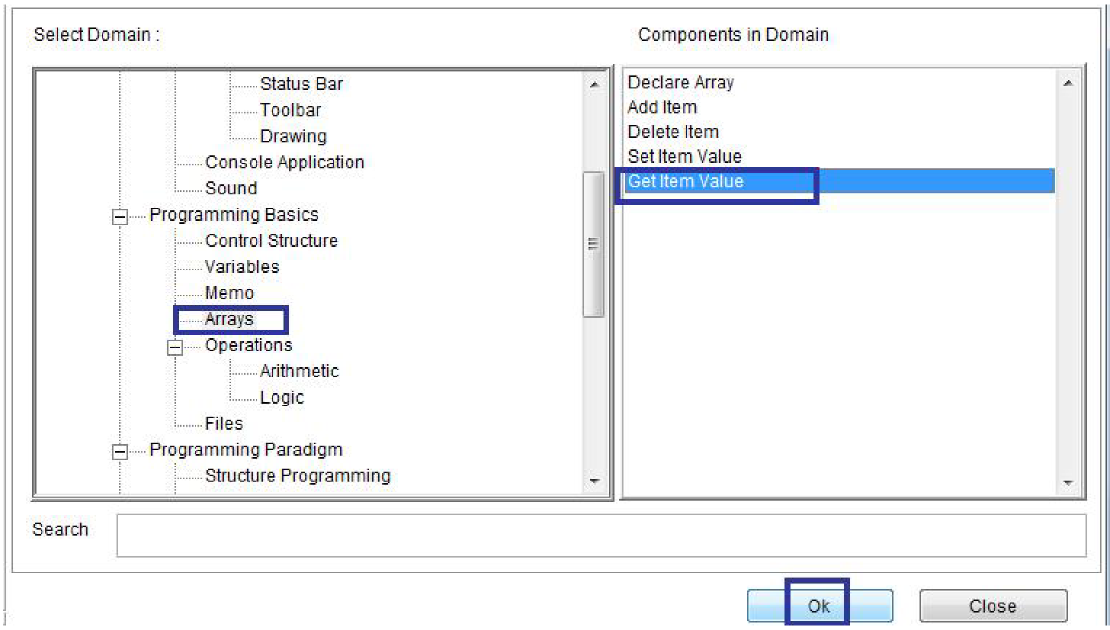

Interaction Page



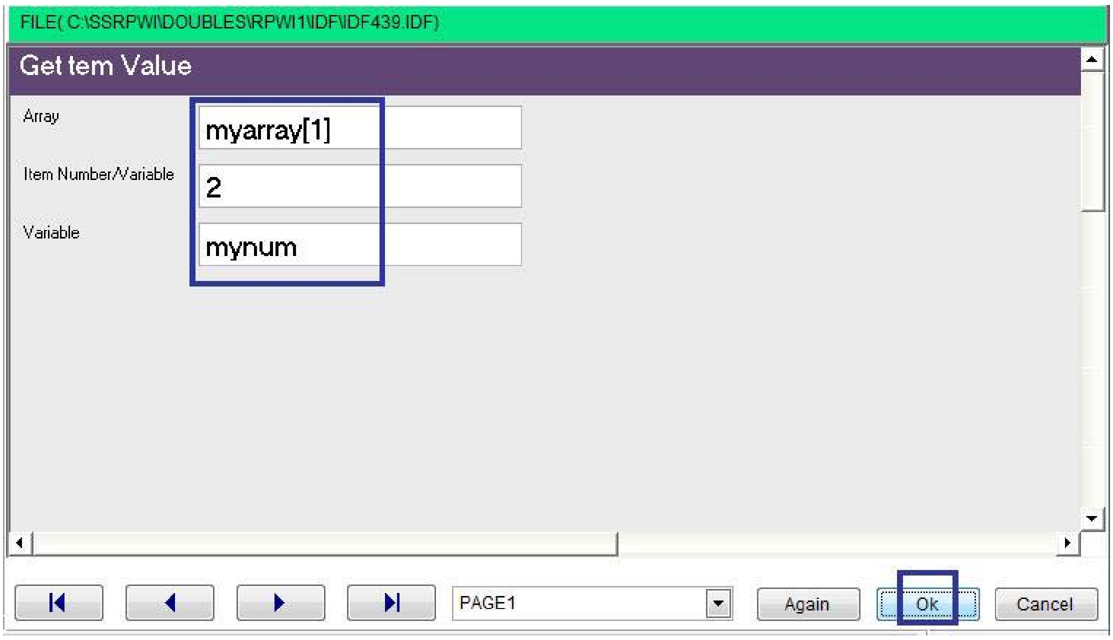

Domain (Arrays) Component (Set Item Value)



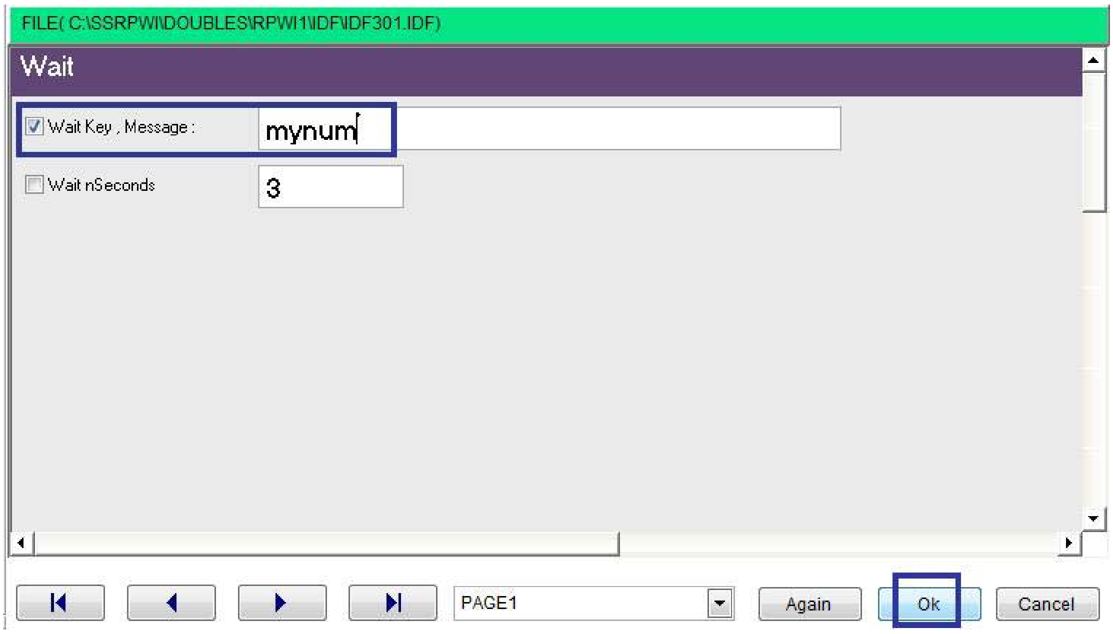

Interaction Page



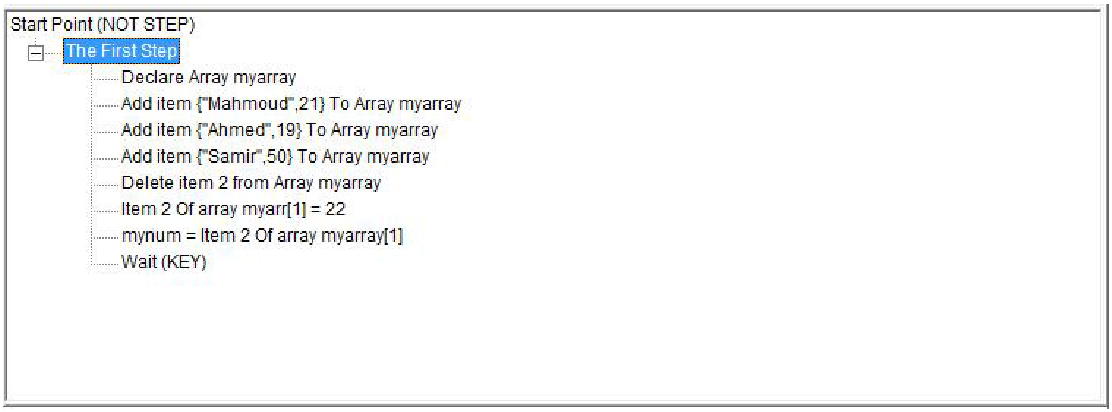

Domain (Arrays) Component (Get Item Value)



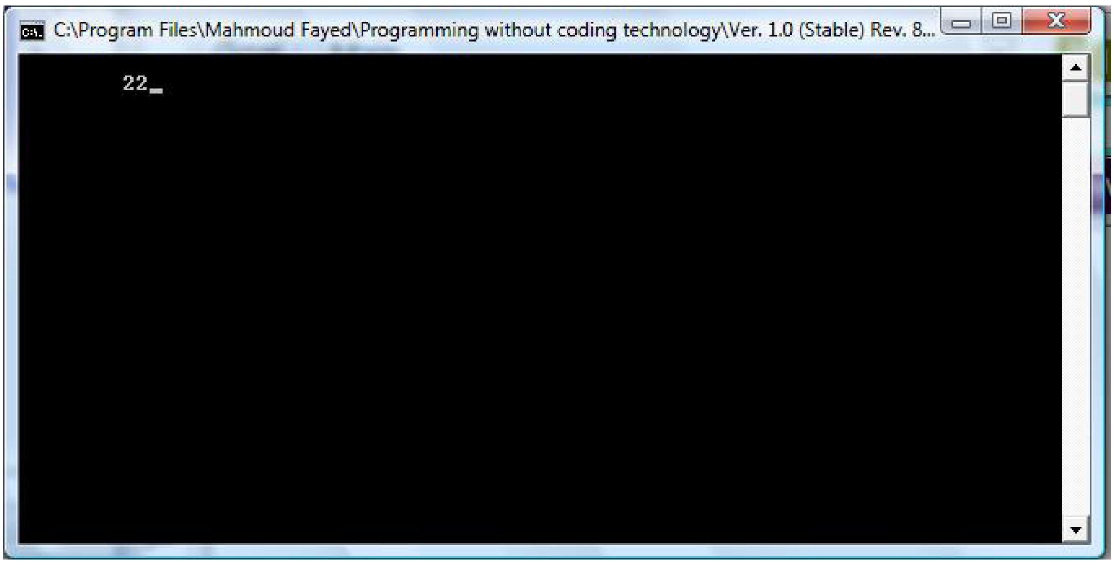

Interaction Page



Interaction Page



Steps Tree



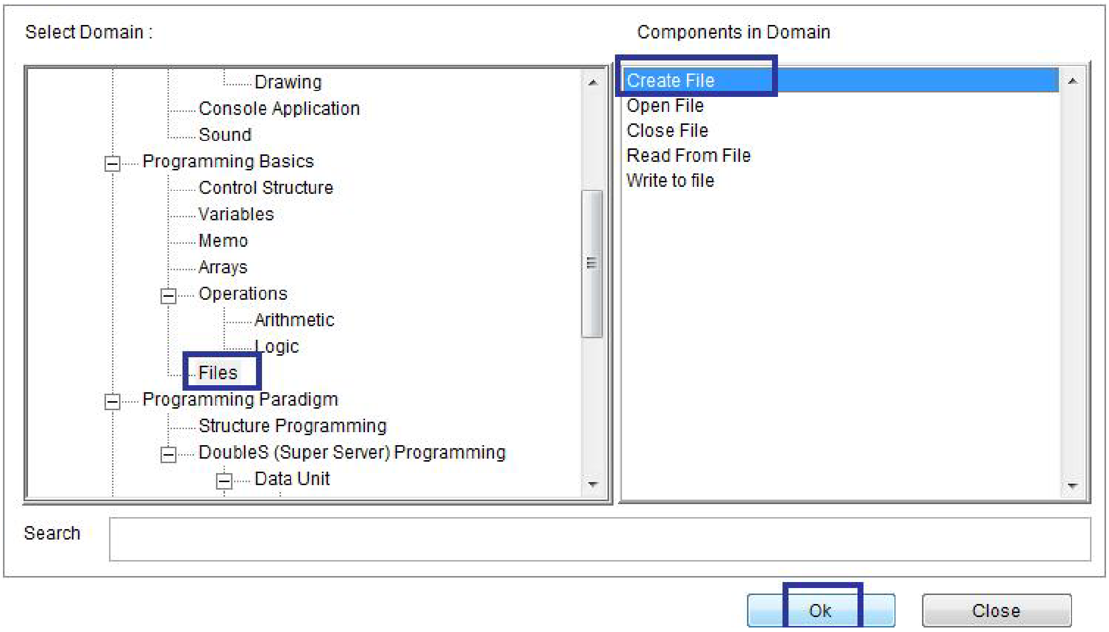
The Final Application

# Files

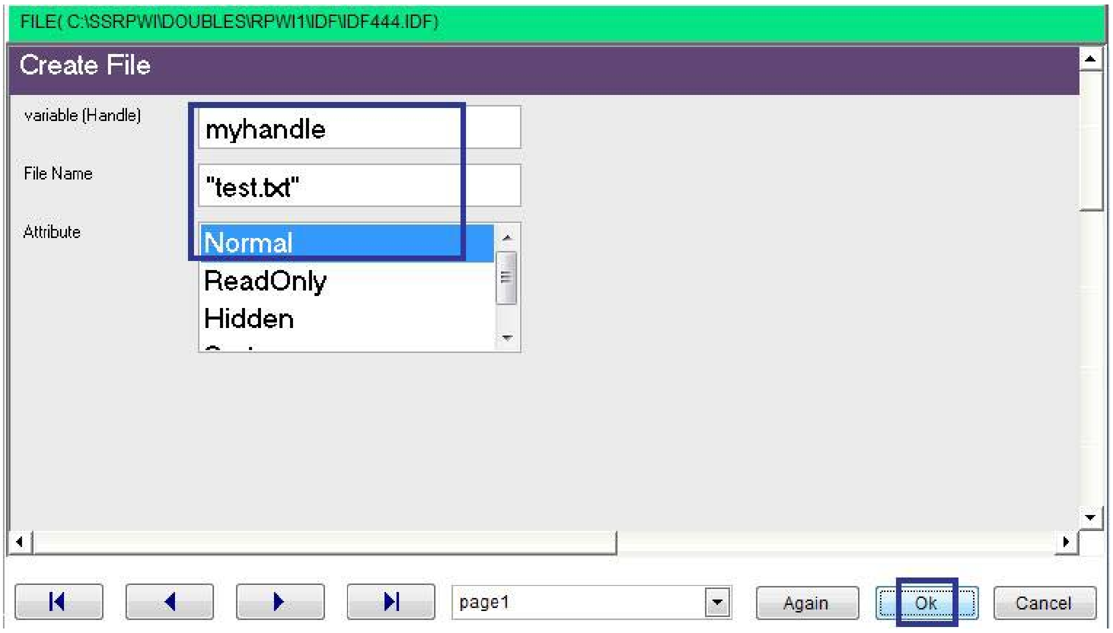
Components

- Create file
- Open file
- Close file
- Read from file
- Write to file

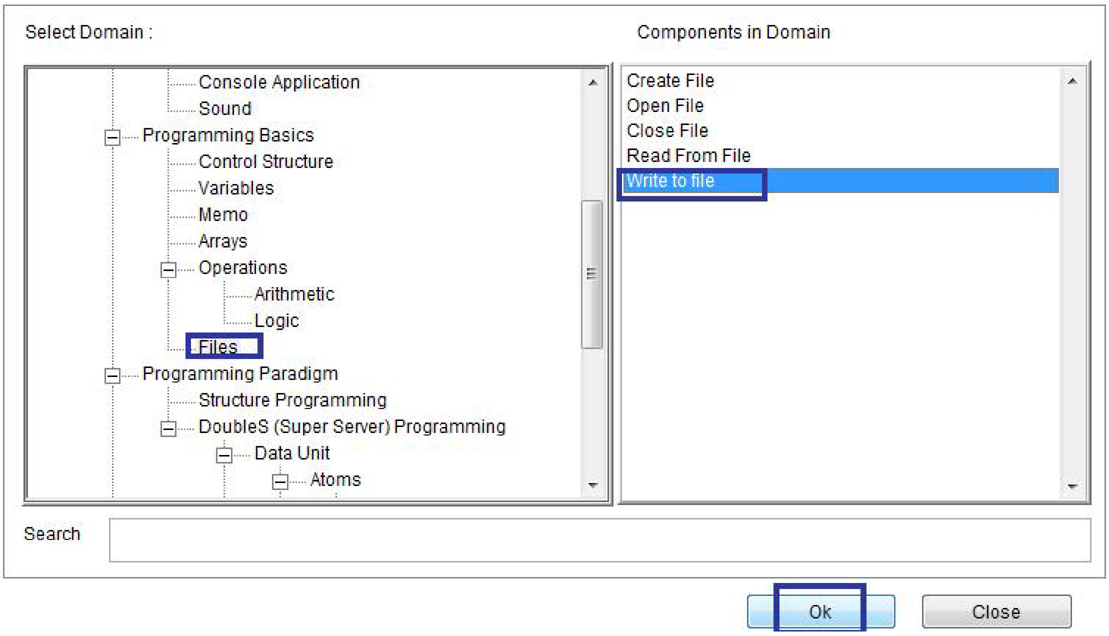
## Example - Screen shots:-



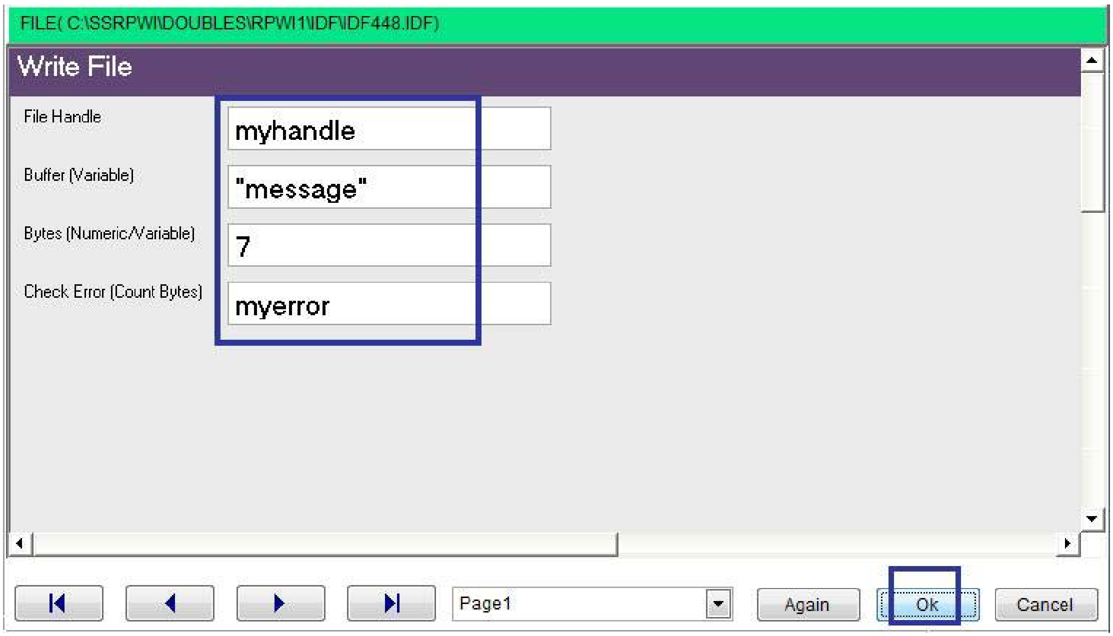

Domain (Files) Component (Create File)



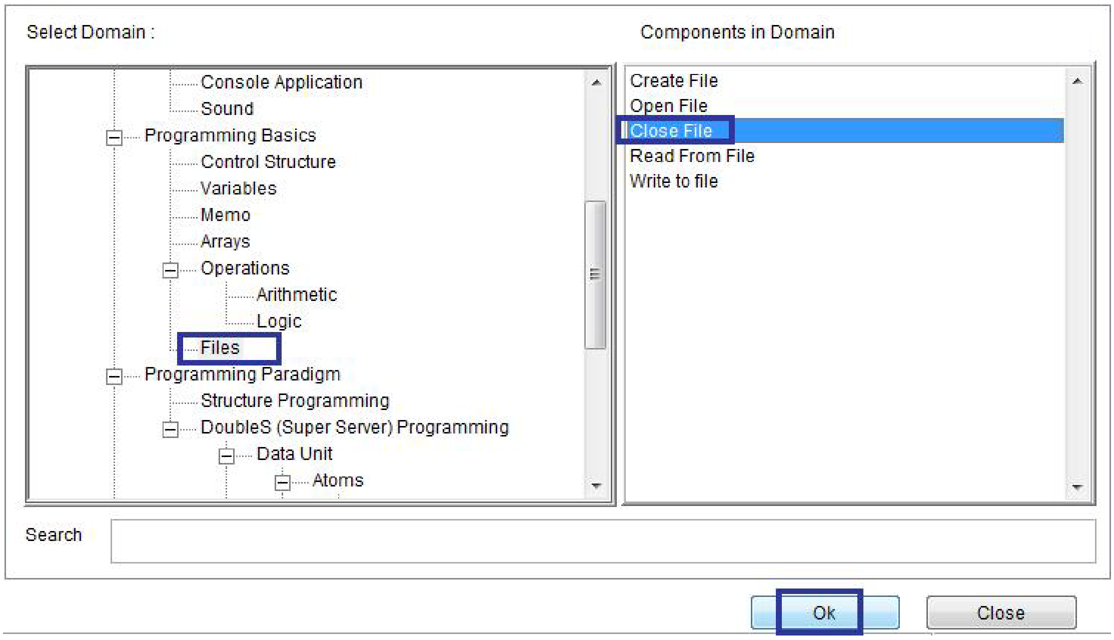

Interaction Page



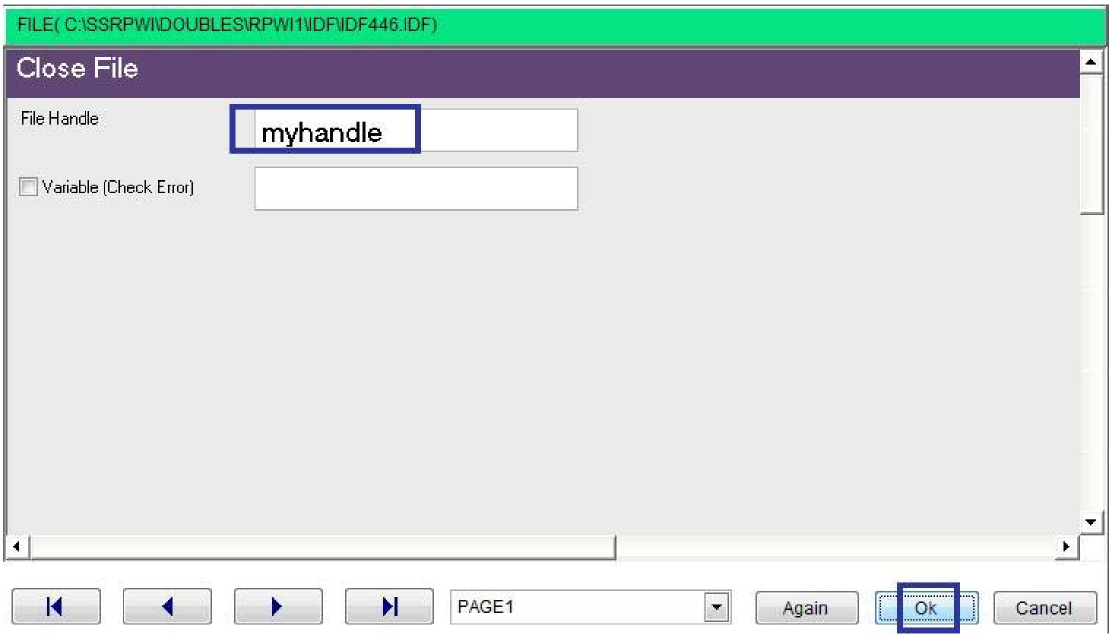

Domain (Files) Component (Write to file)



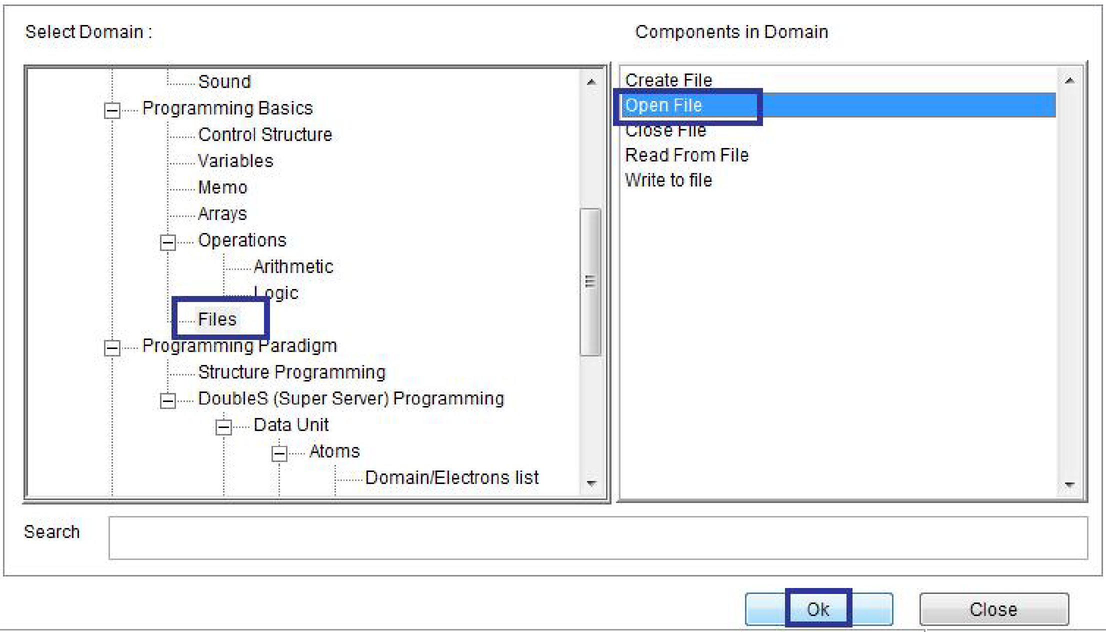

Interaction Page



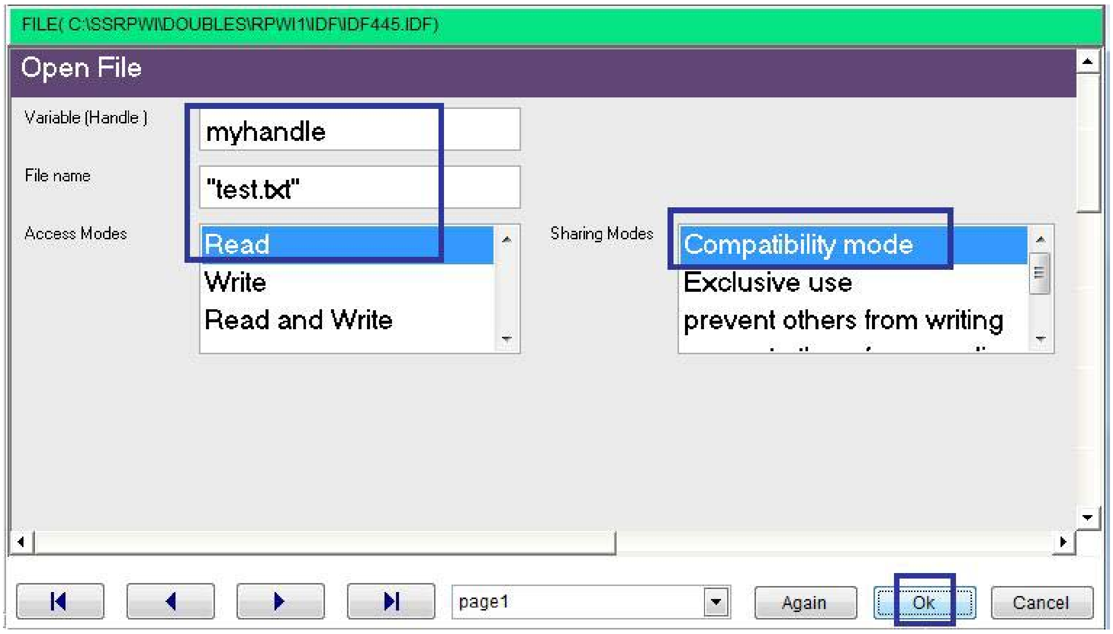

Domain (Files) Component (Close File)



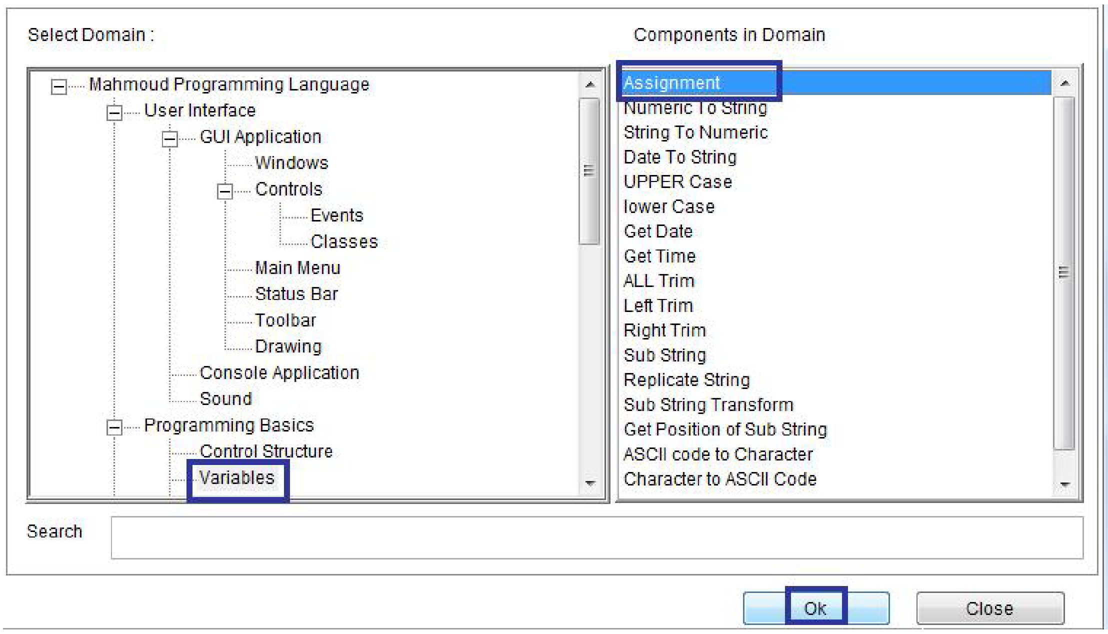

Interaction Page



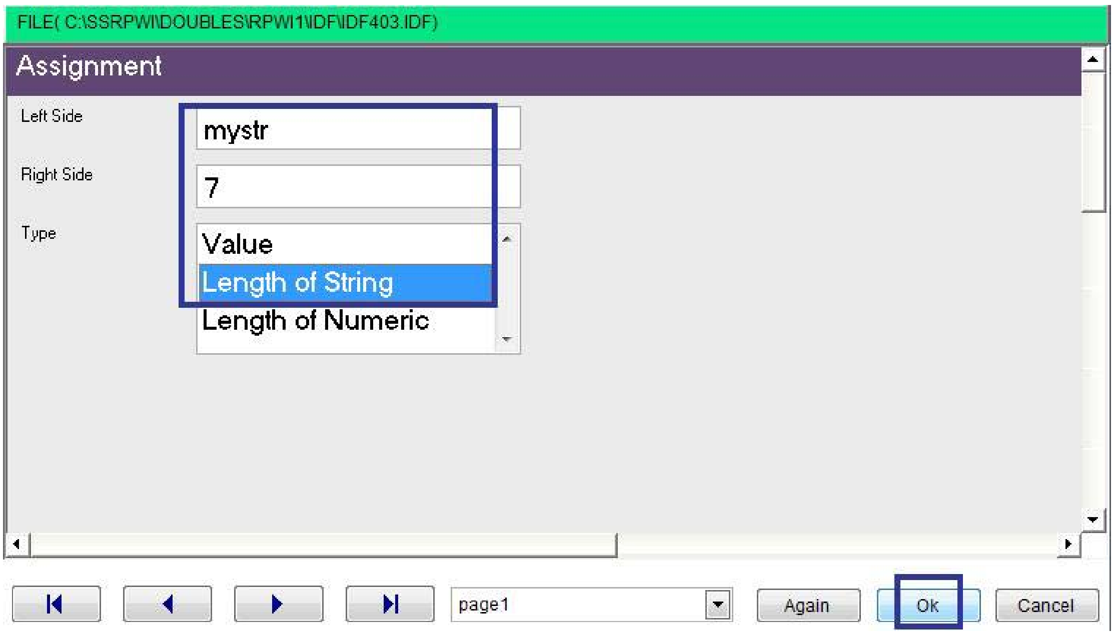

Domain (Files) Component (Open File)



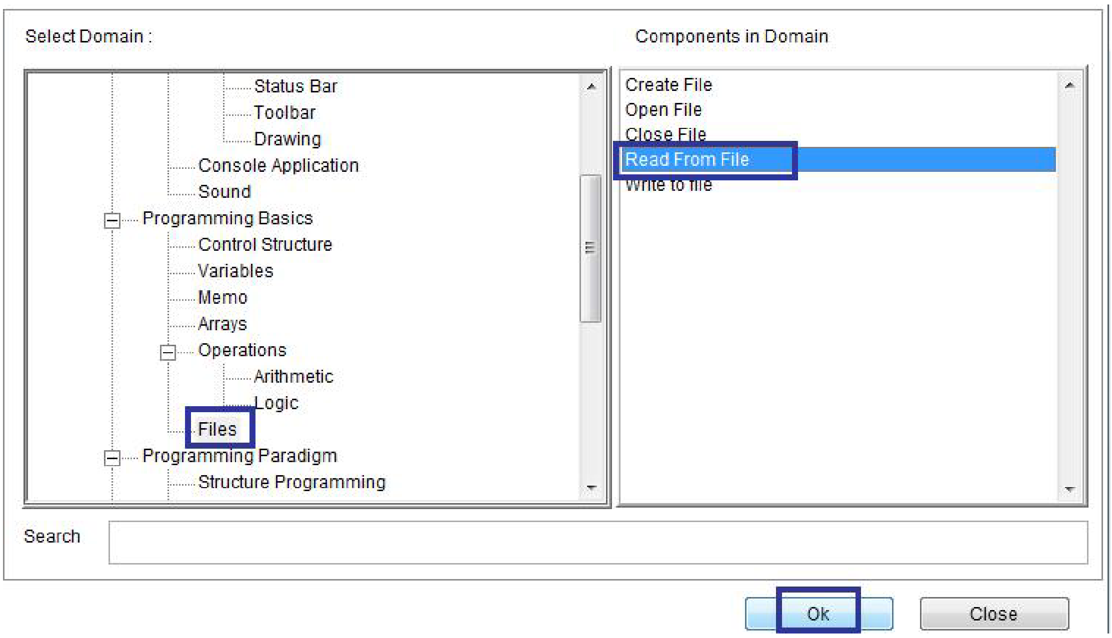

Interaction Page



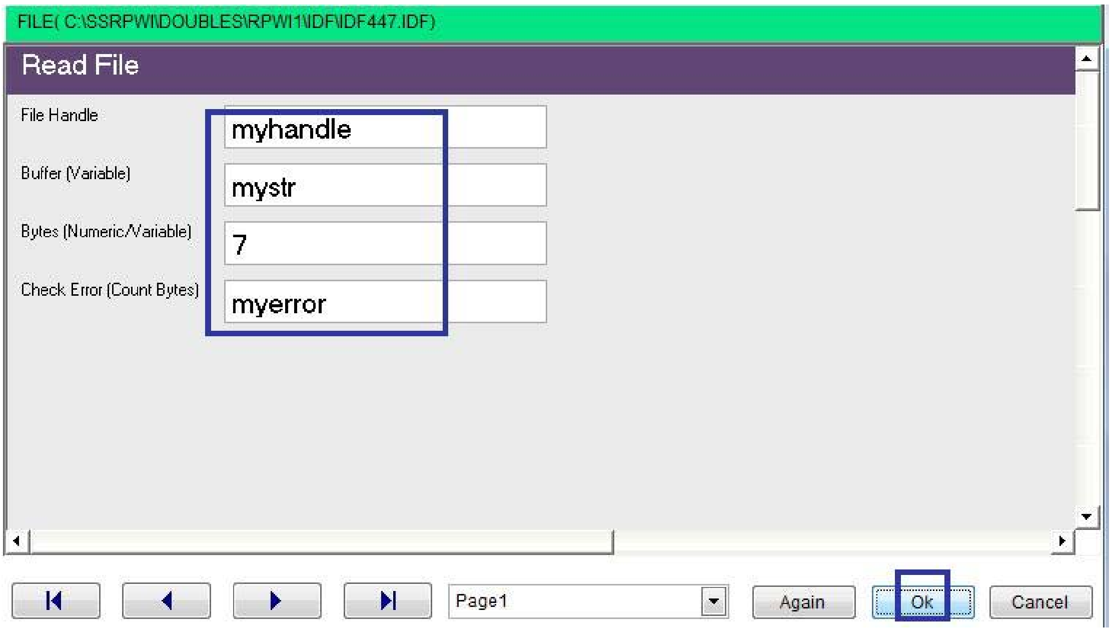

Domain (Variables) Component (Assignment)




Interaction Page

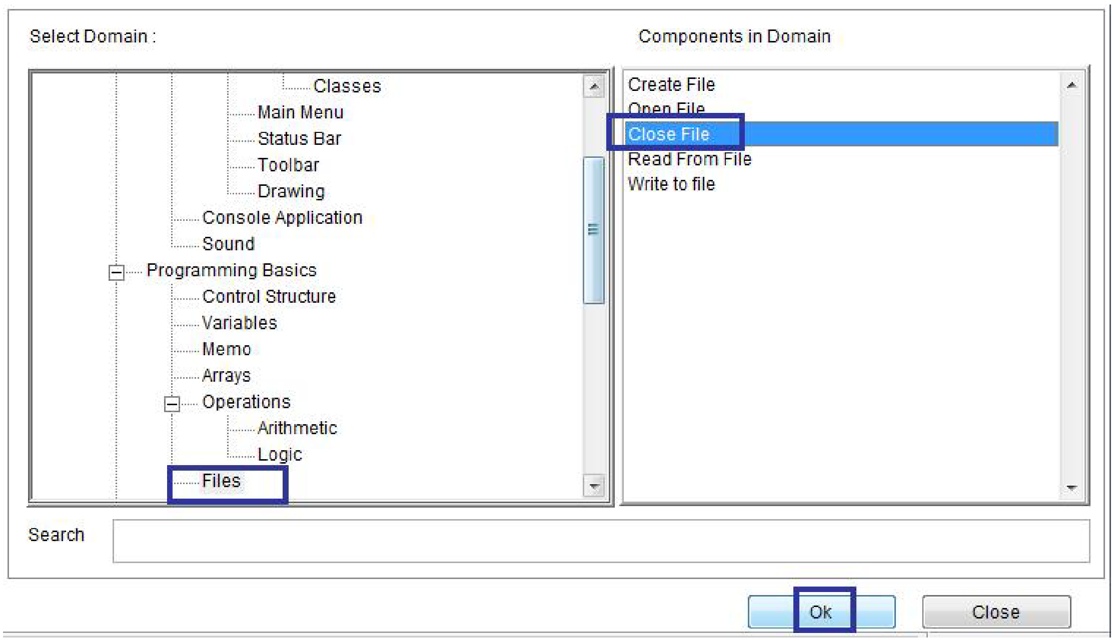



Domain (Files) Component (Read from File)




Interaction Page

`

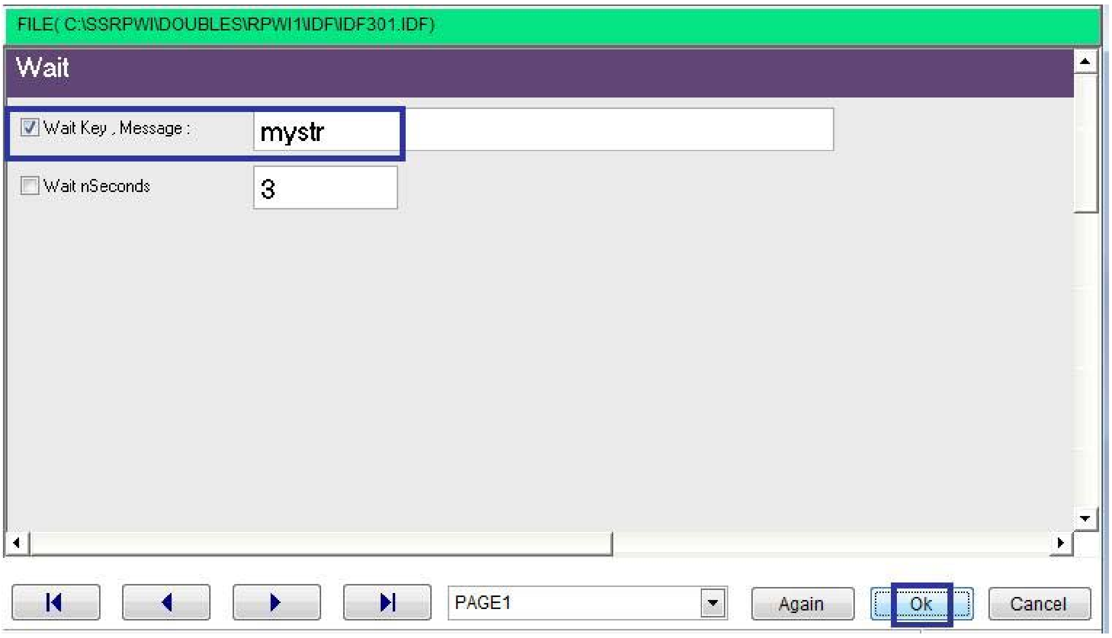



Domain (Files) Component (Close File)

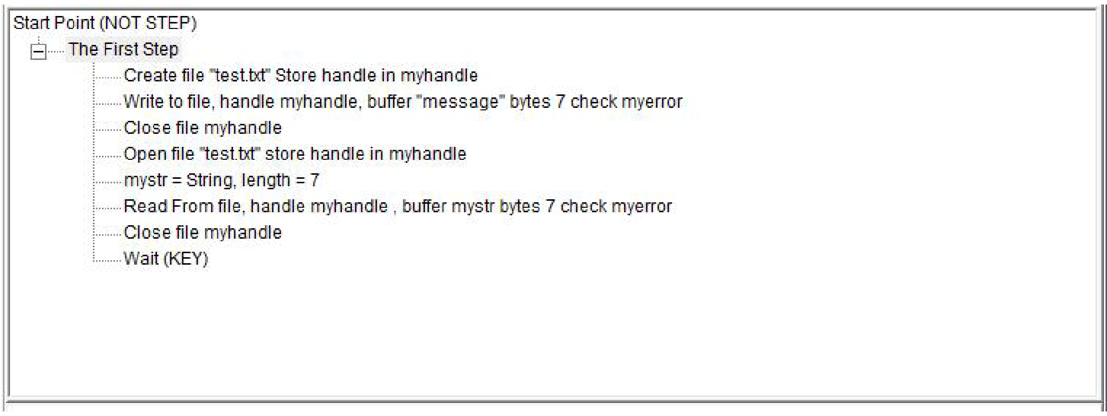



Interaction Page

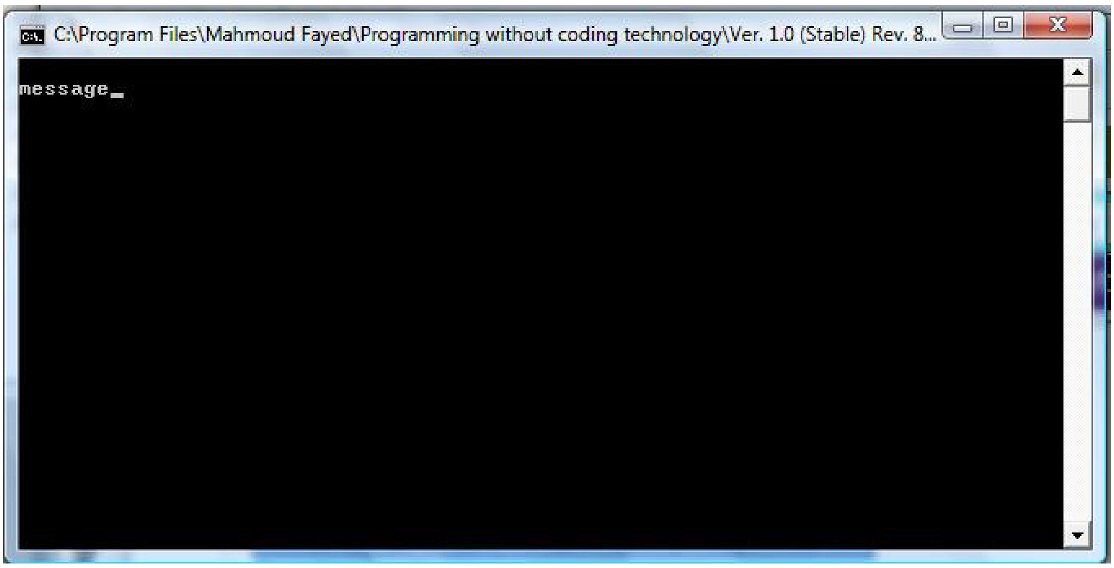
`




Interaction Page




Final Steps Tree




Final Application

`

# Structure Programming

Components

- Define Procedure
- Define Function
- Scope of variables
- Return Value
- Call Procedure
- Call Function
- Release Variable

Procedure may be a generated server file (*.prg) or inline procedure.
Procedure does not return a value.

Scope of variables:

Variables may have the following scope:

LOCAL: Visible only within the routine which declared it. Value is lost upon exit of the routine.

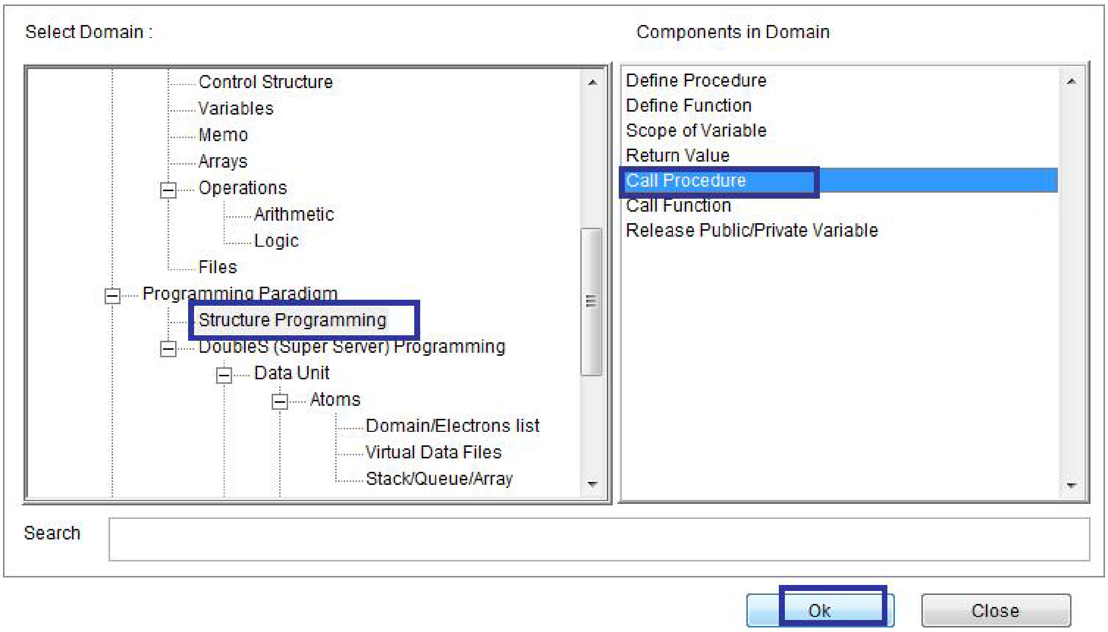
STATIC: Visible only within the routine which declared it. Value is preserved for subsequent invocations of the routine. If a STATIC variable is declared before any Procedure/Function/Method is defined, it has a MODULE scope, and is visible within any routine defined within that same source file, it will maintain its life for the duration of the application life time.

PRIVATE: Visible within the routine which declared it, and all routines called by that routine.

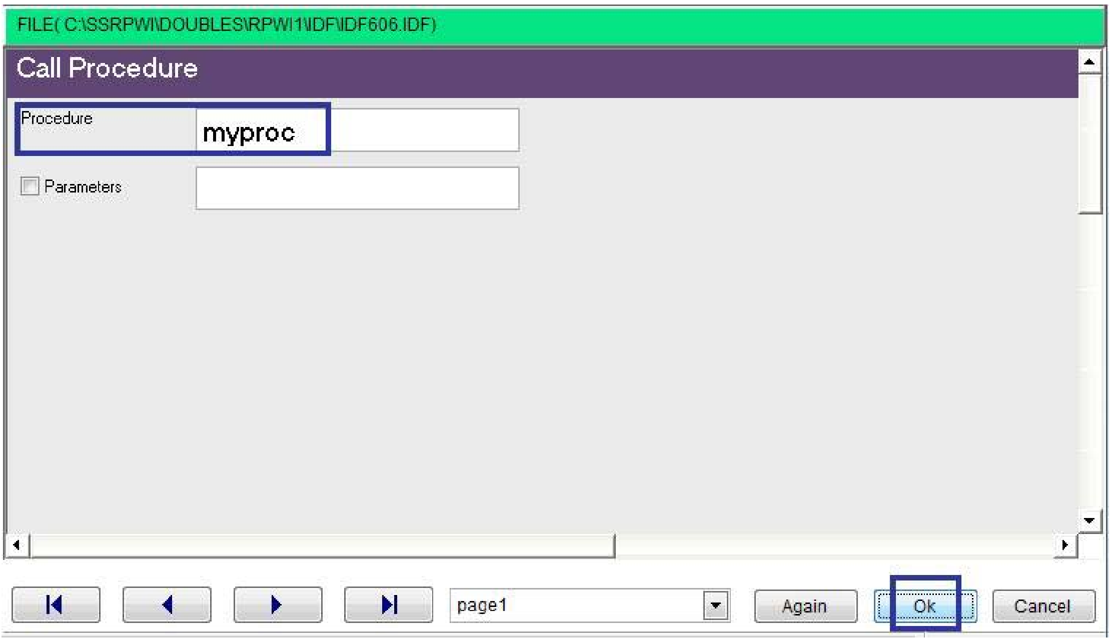
PUBLIC: Visible by all routines in the same application.

Due to the dynamic nature of PRIVATE and PUBLIC variables, they can be created and destroyed at Run-time

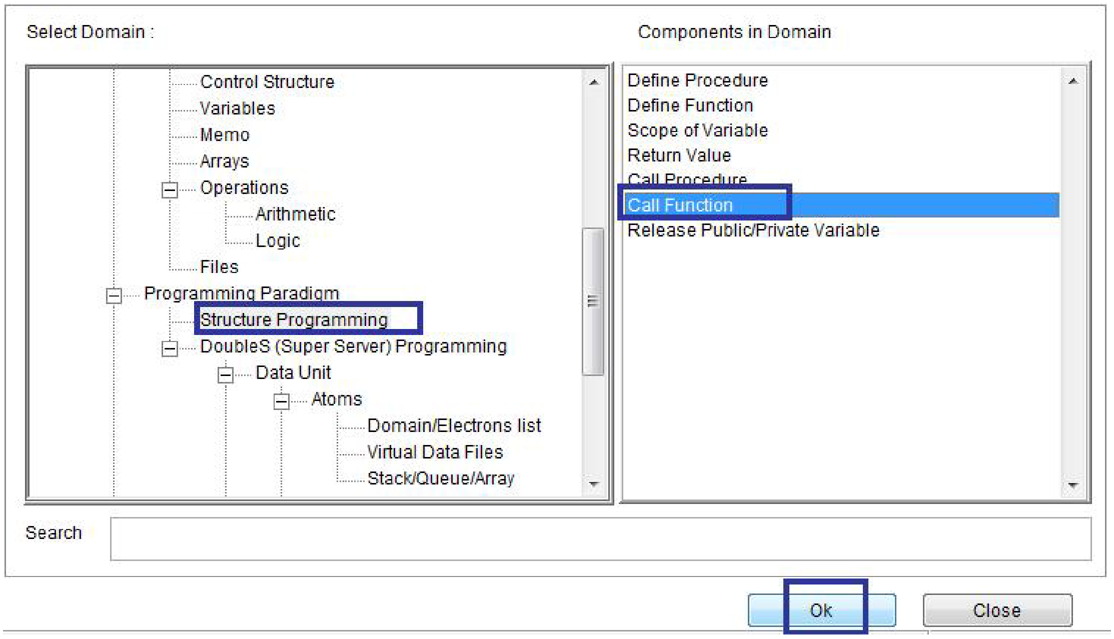
## Example - Screen shots:-



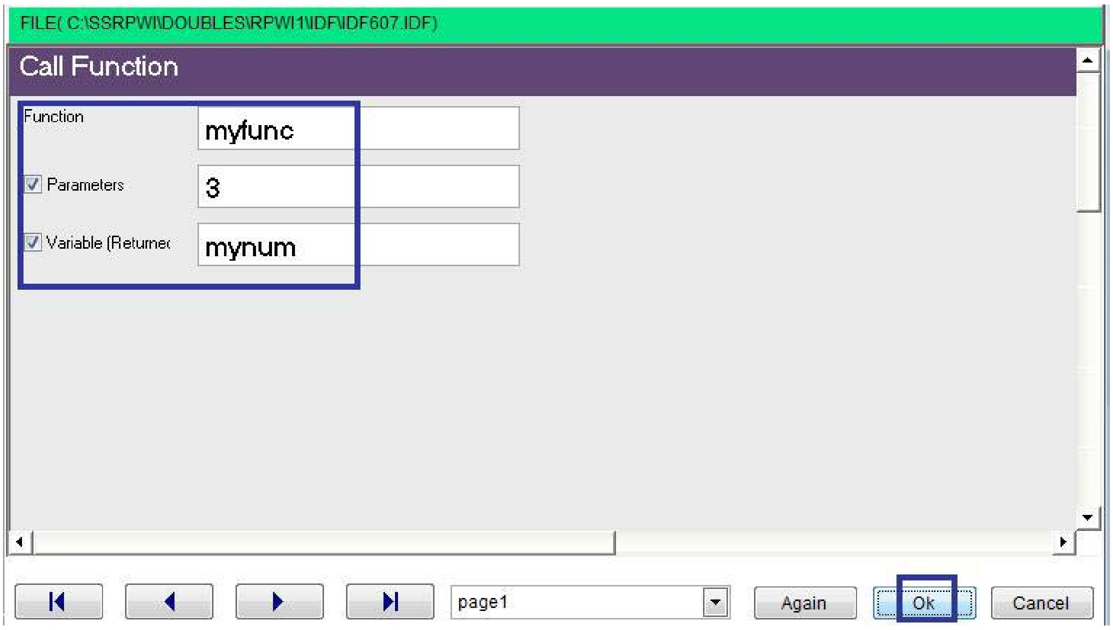

Domain (Structure Programming) Component (Call Procedure)



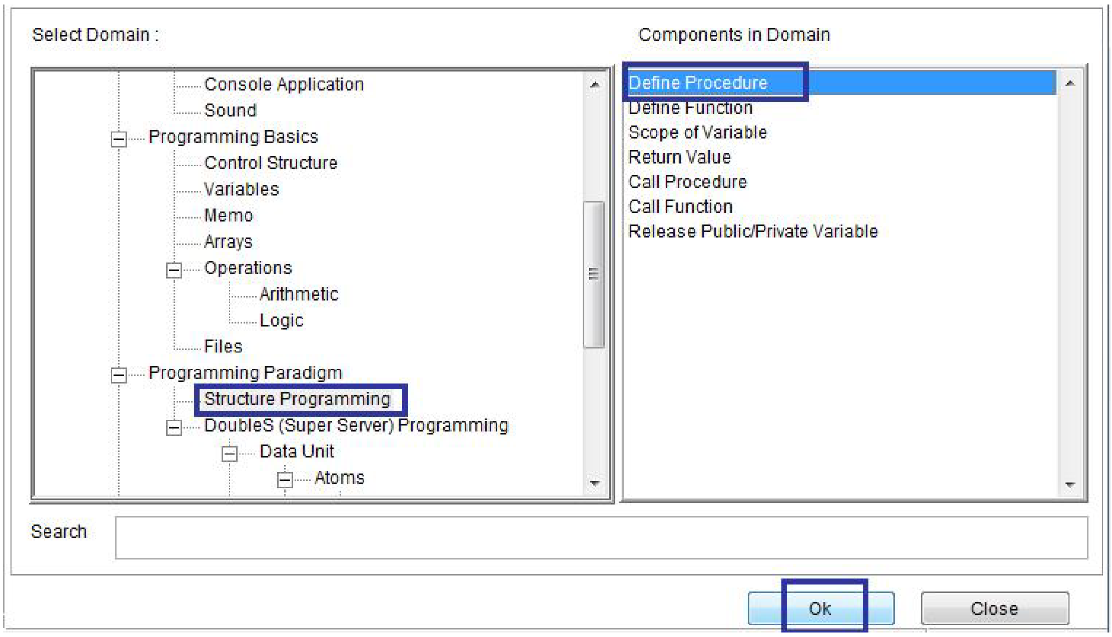

Interaction Page



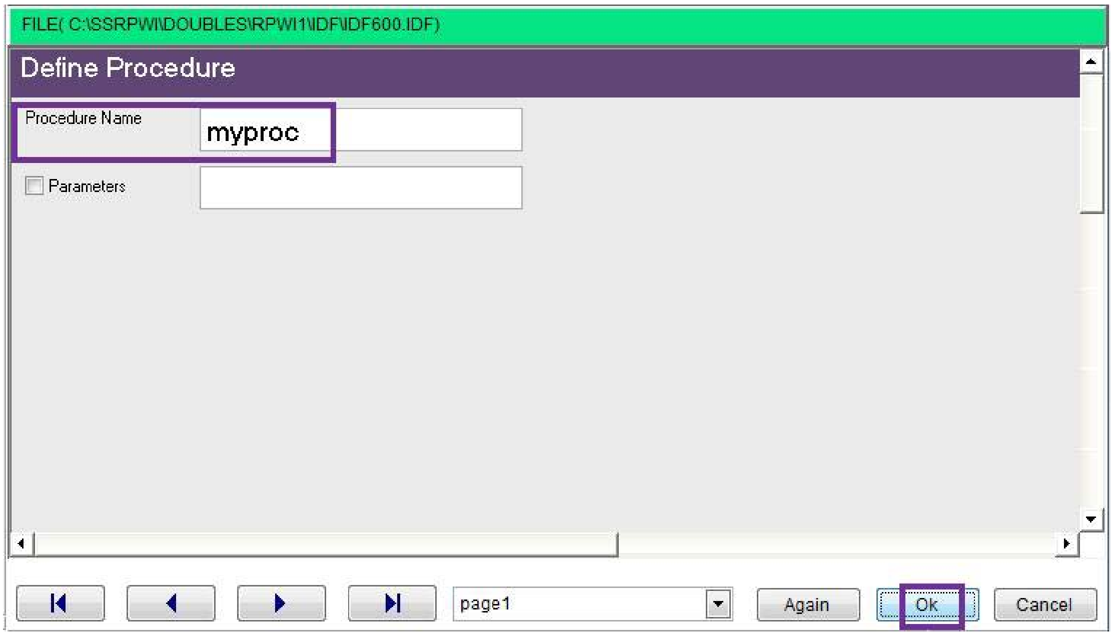

Domain (Structure Programming) Component (Call Function)



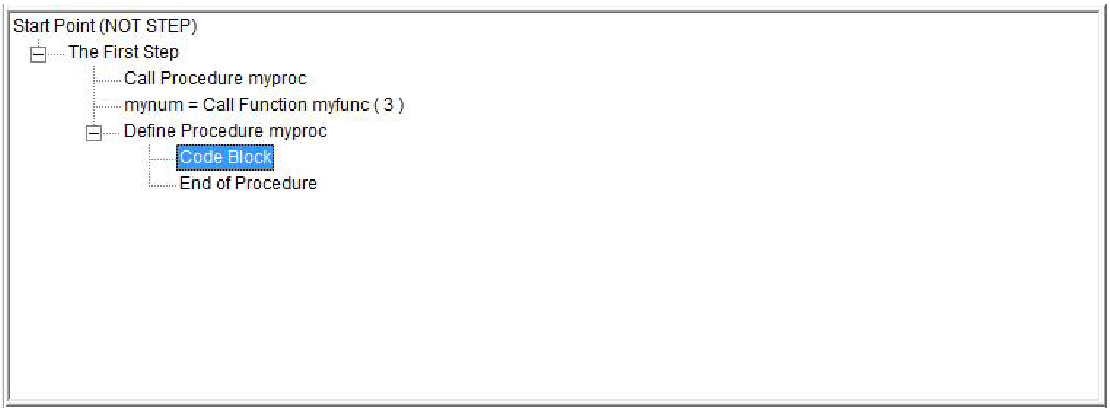

Interaction Page



Domain (Structure Programming) Component (Define Procedure)

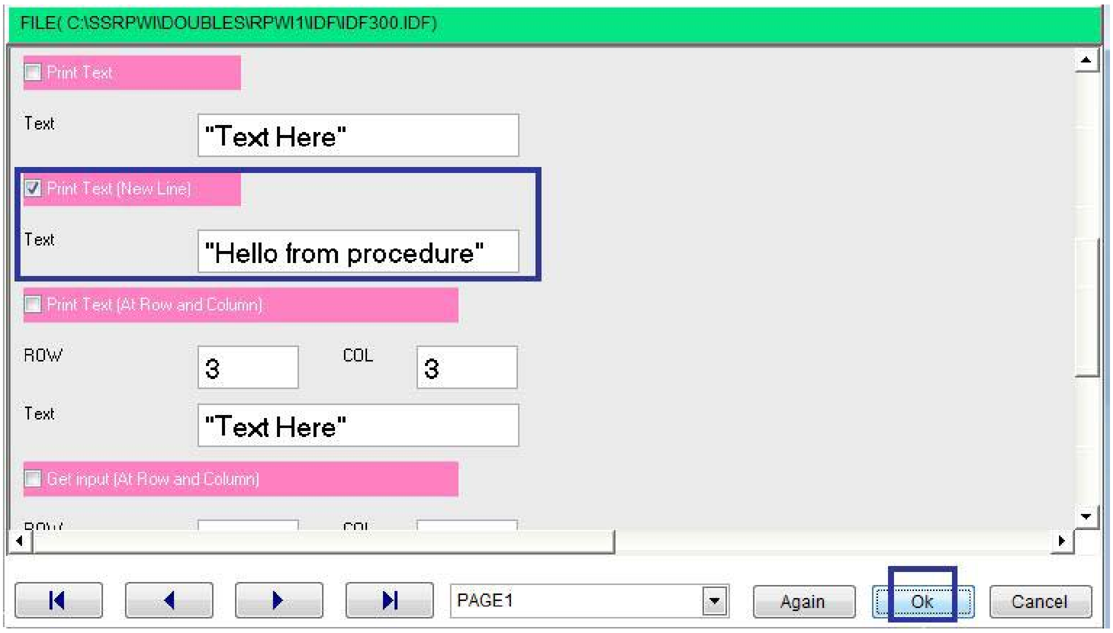


Interaction Page

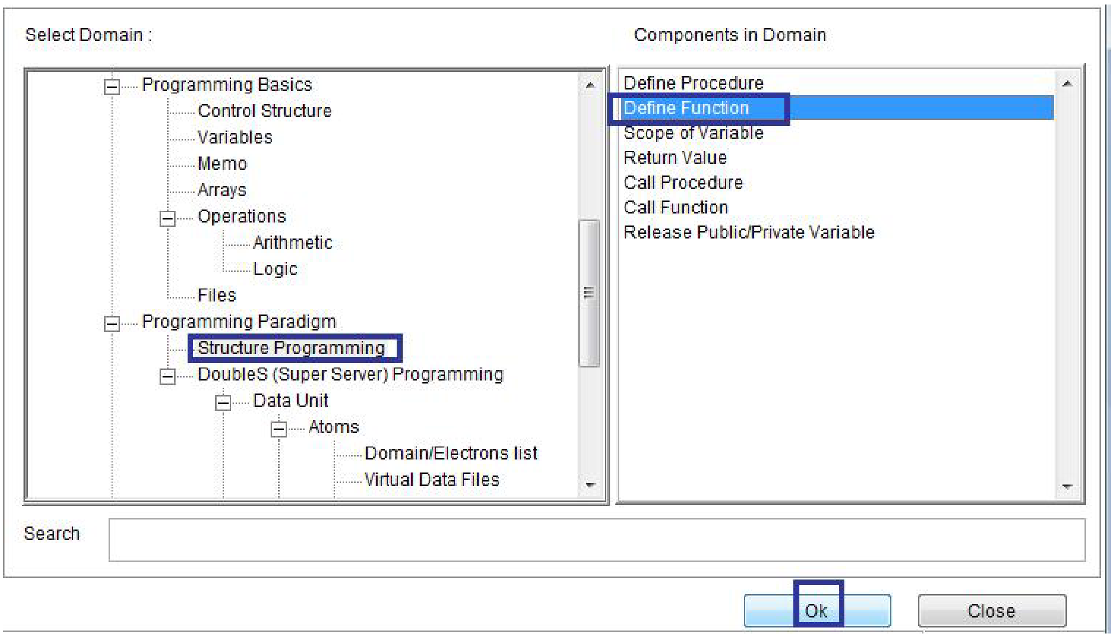


Steps Tree

`

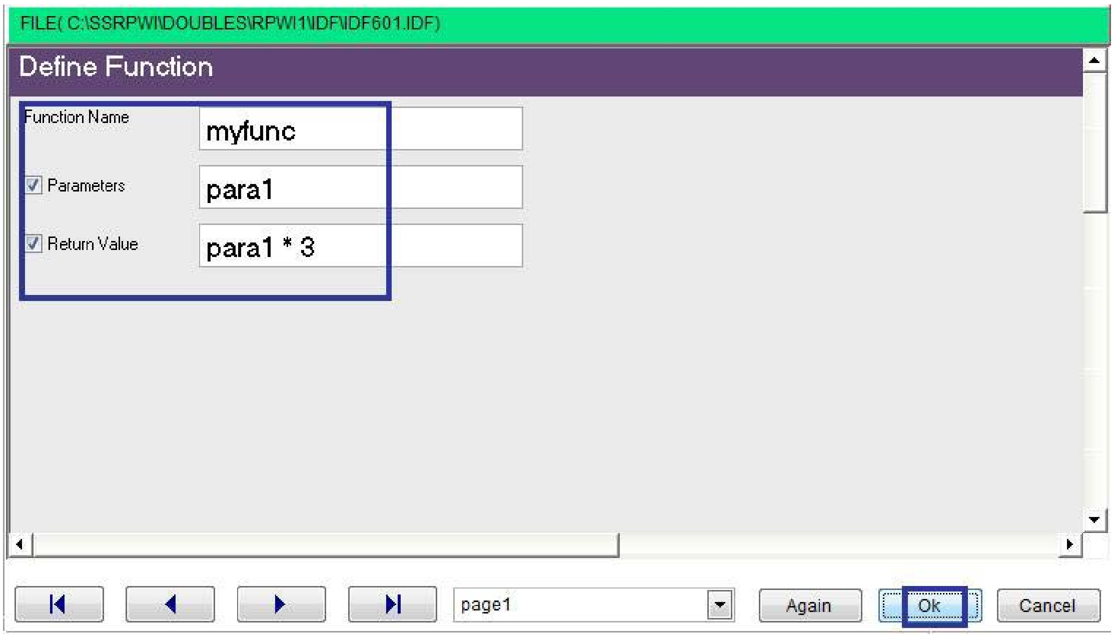



Interaction Page

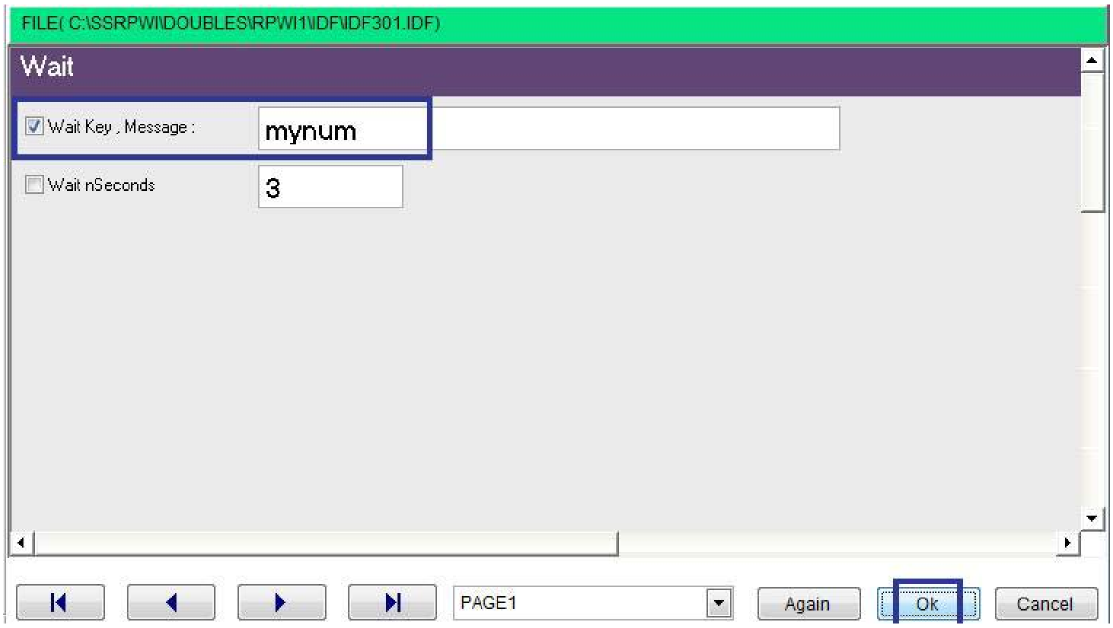



Domain (Structure Programming) Component (Define Function)

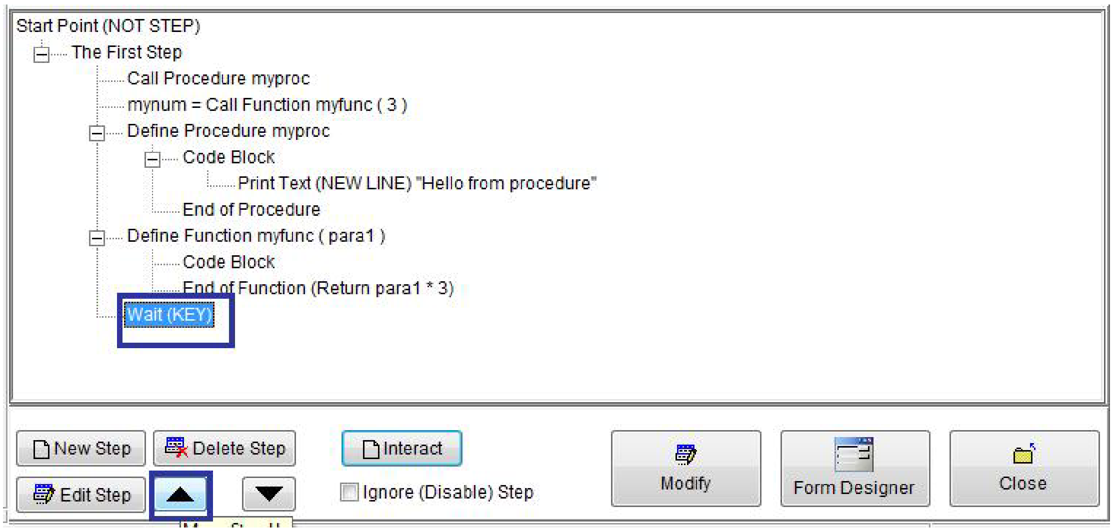
`



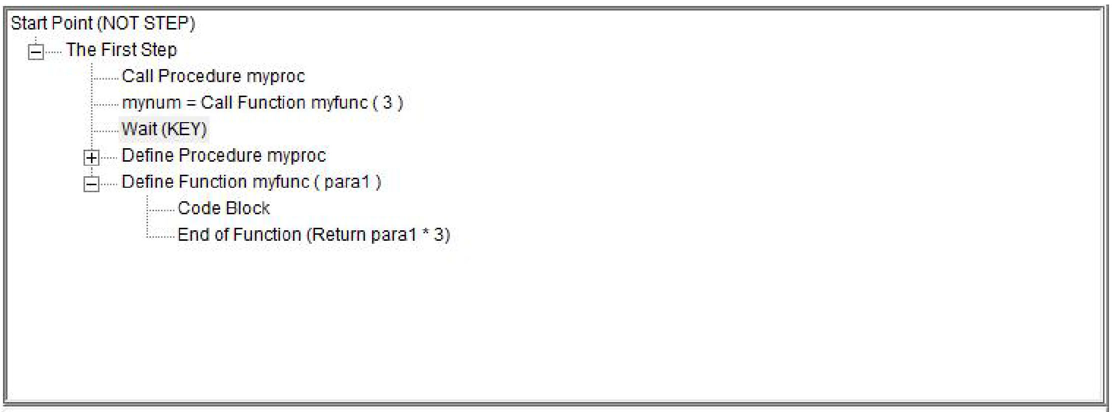

Interaction Page



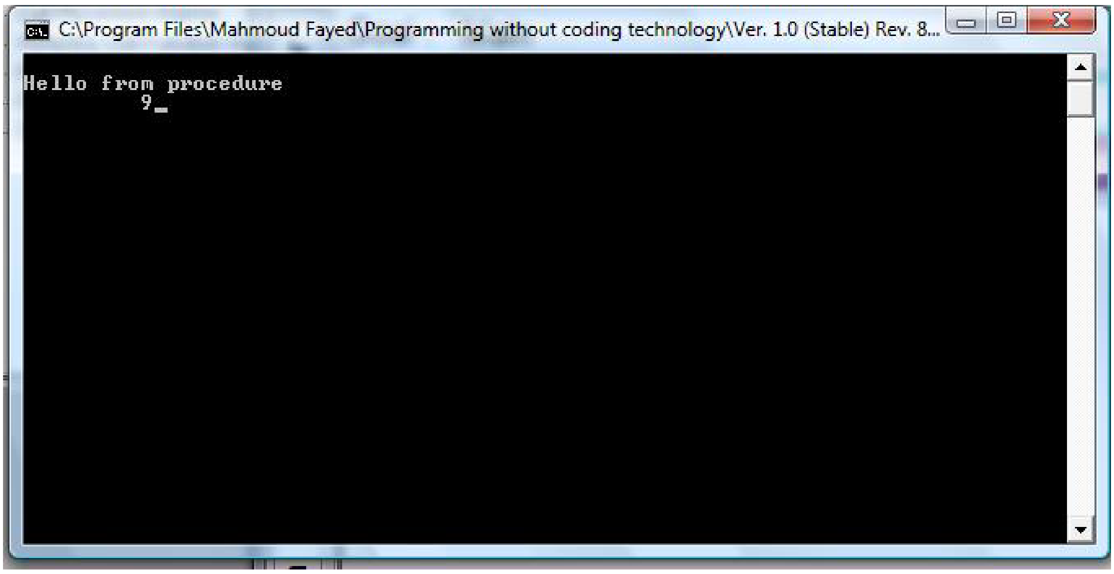

Interaction Page




Steps Tree




Final Steps Tree




Final Application

# Database Files

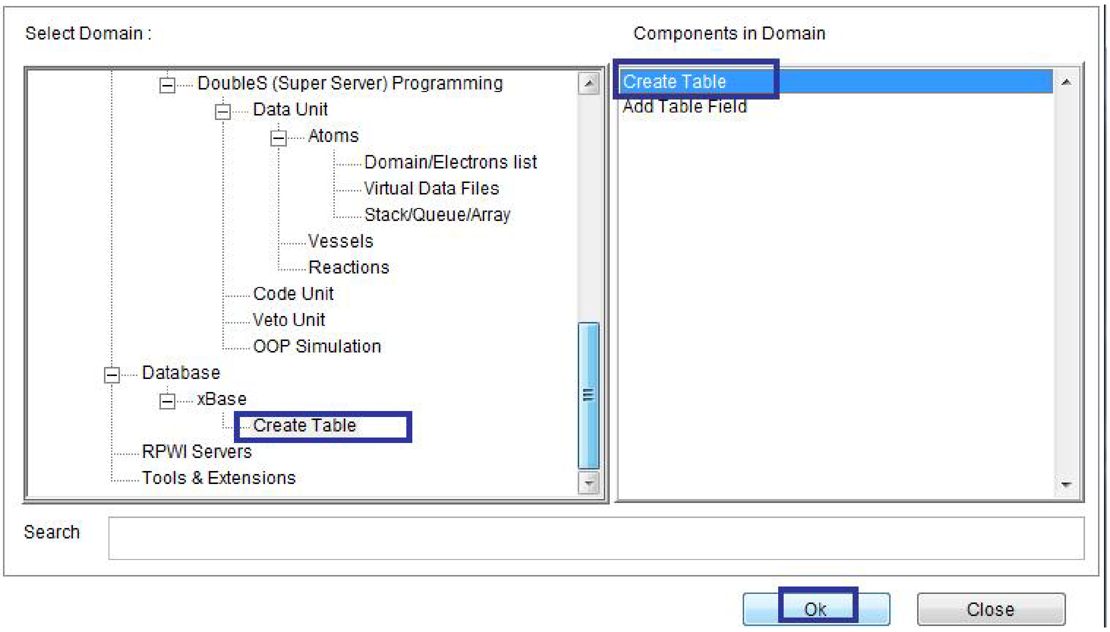
- Create Table
- Opening & Closing Tables
- Add new record
- Moving between records
- Record number & records count
- Modify record
- Deleting records
- Search
- Index Files
- Shared Data

## Create Table

Table Name: TEL.DBF (Data Base File)

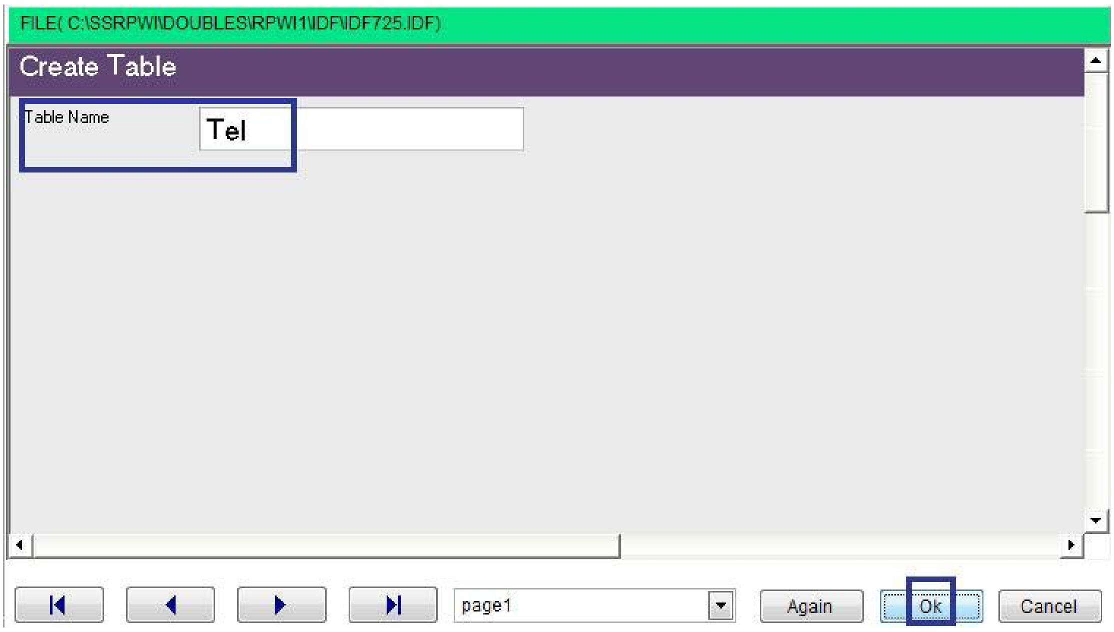
| Field | Type | Size |
|---|---|---|
| Name | Character | 50 |
| Address | Character | 50 |
| Phone | Character | 15 |

### Example - Screen shots:-




Domain(Create Table) Component (Create Table)

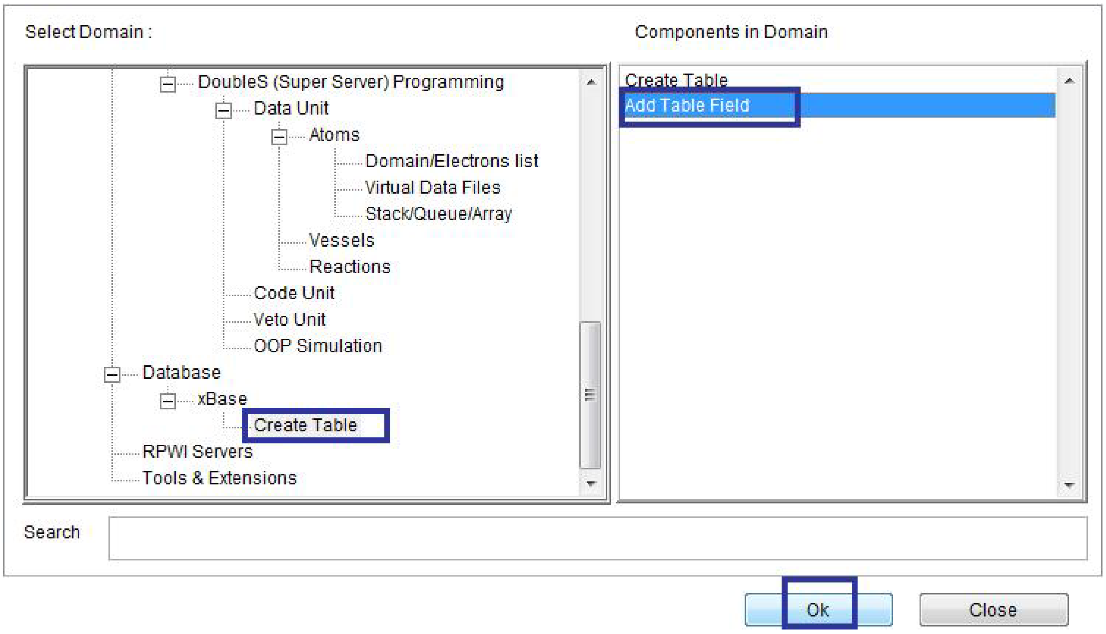
`



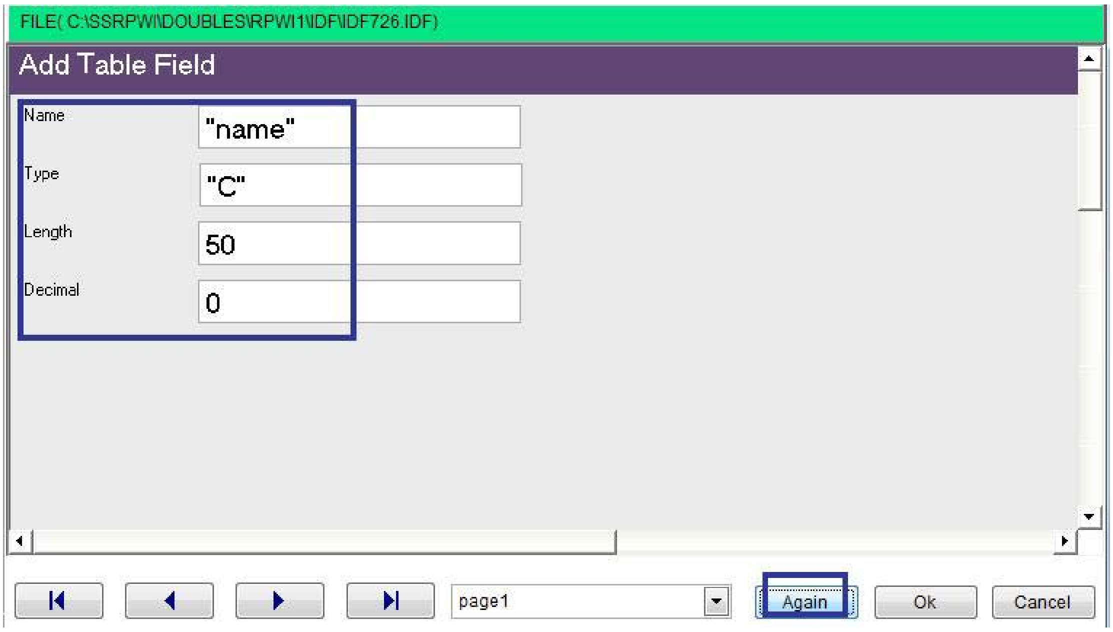

Interaction Page



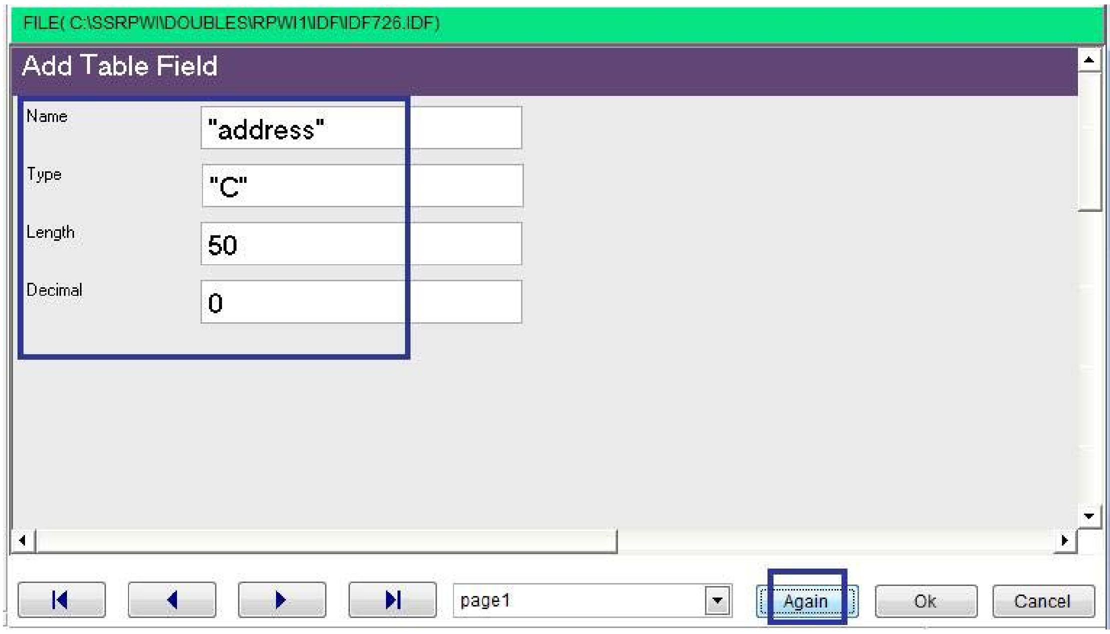

Steps Tree



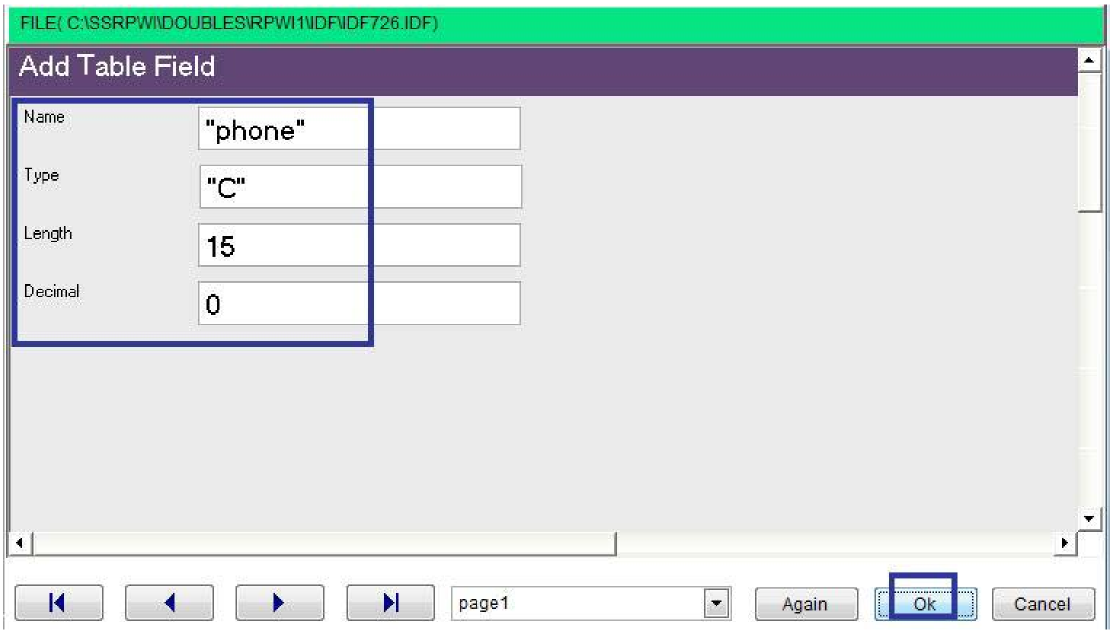

Domain (Create Table) Component (Add Table Field)



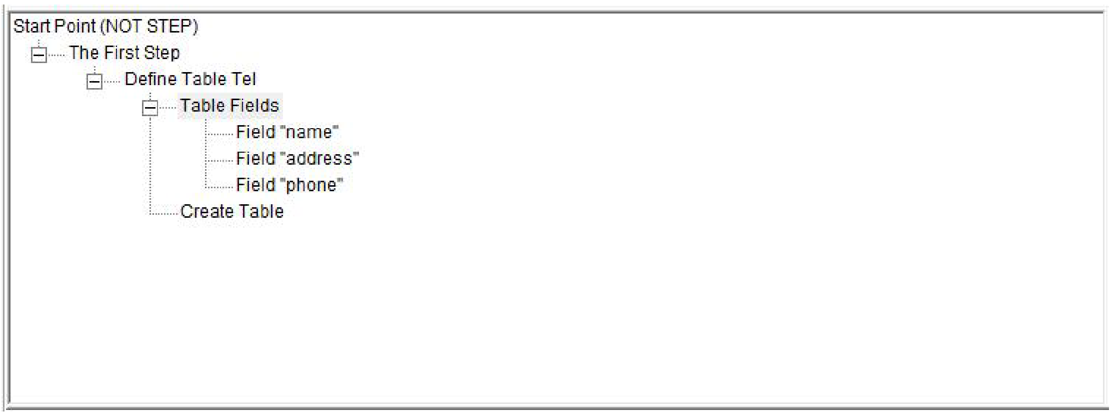

Interaction Page




Interaction Page




Interaction Page




Final Steps Tree

`

## Opening & Closing Tables

Work Area are used to host a table, you can open more than one table in the same time (up to 250) using work areas.

What you need to do for example

- Select work area 1
- Open  Table1
- Select work area 2
- Open  Table2
- ……………….
- ……………….
- Select work area n
- Open TableN

Note that the work area number is not related to the table name
Where you are free to determine what work area host what table

To change the active area (moving between tables), all what you need is to select the work are of the table which you want to deal with

The table in the active work area (selected area) is under your usage directly
Where you can use its field's names as variable names to get their content (Reading) or altering their content (writing)

When you open table, the active record is the first record, but you can move between records (First Record, Last Record, Next Record & Previous Record) to get their data or alter it

Also you can know the record number and number of records inside the table

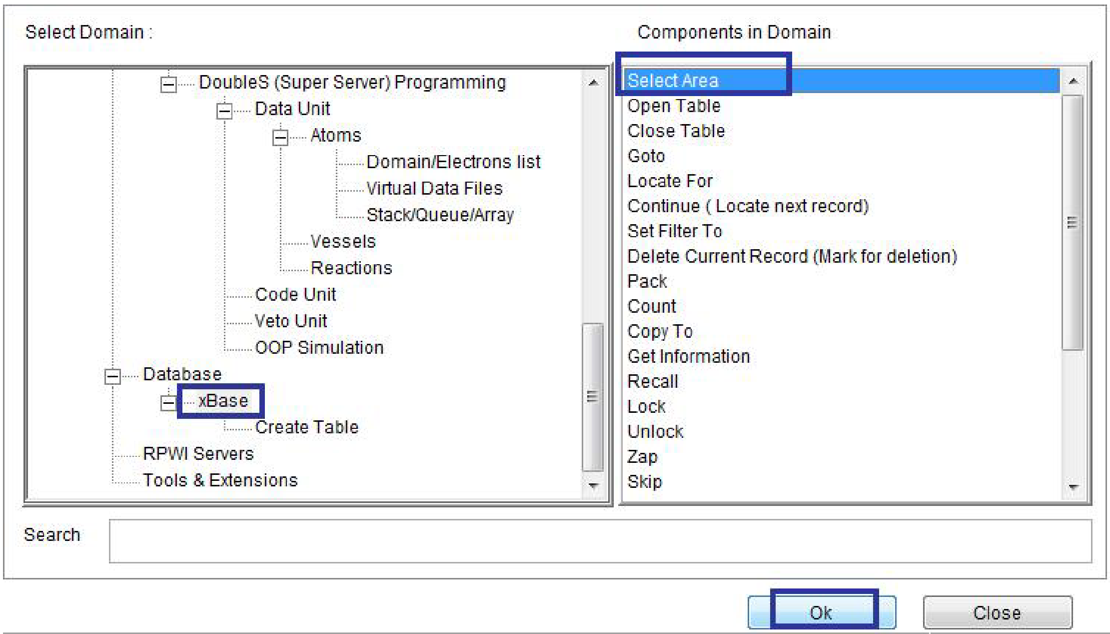
It's clear that working with database files depend on status which include

- Work Area
- Table Name
- Record Number

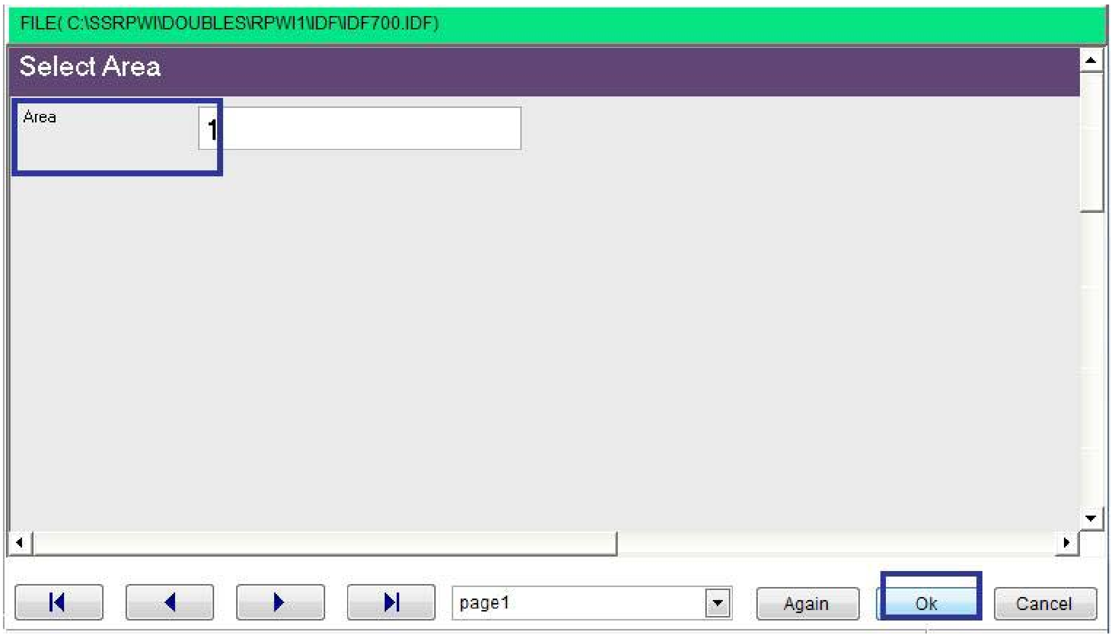
To close a table, you need to activate the work area of this table then close the table
Also you can close all the opened tables (close database)

## Add New Record

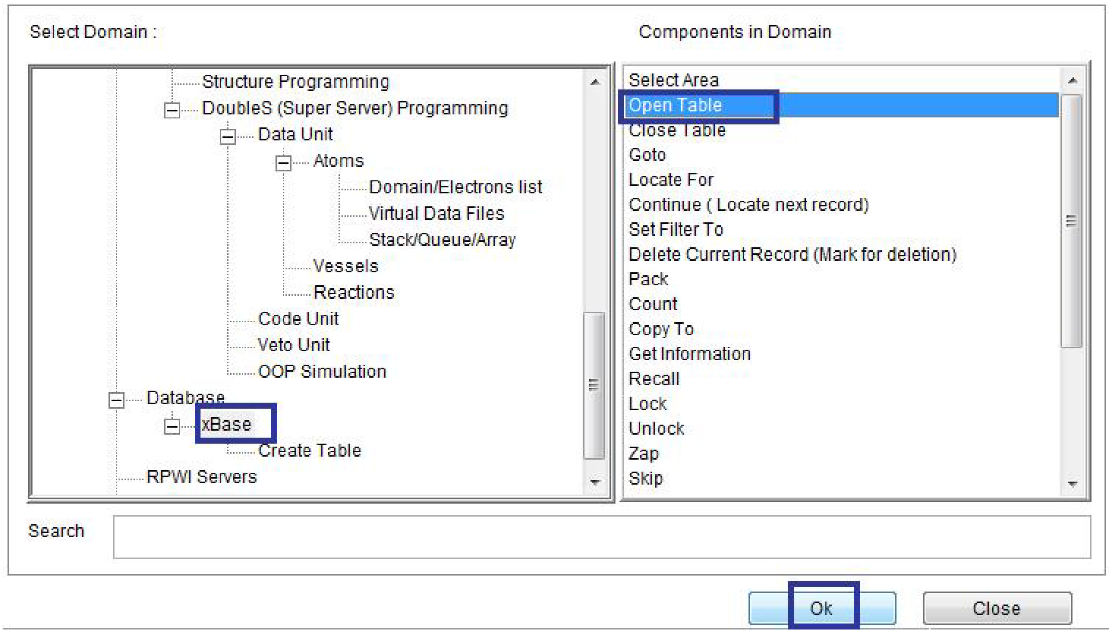
### Example - Screen shots:-



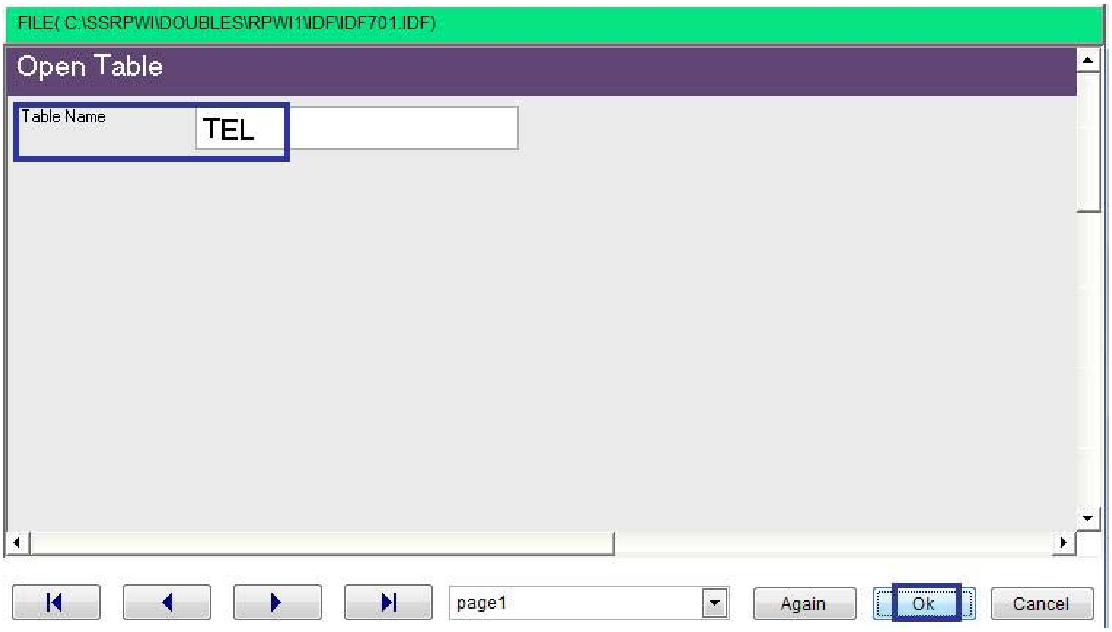

Domain (xBase) Component (Select Area)



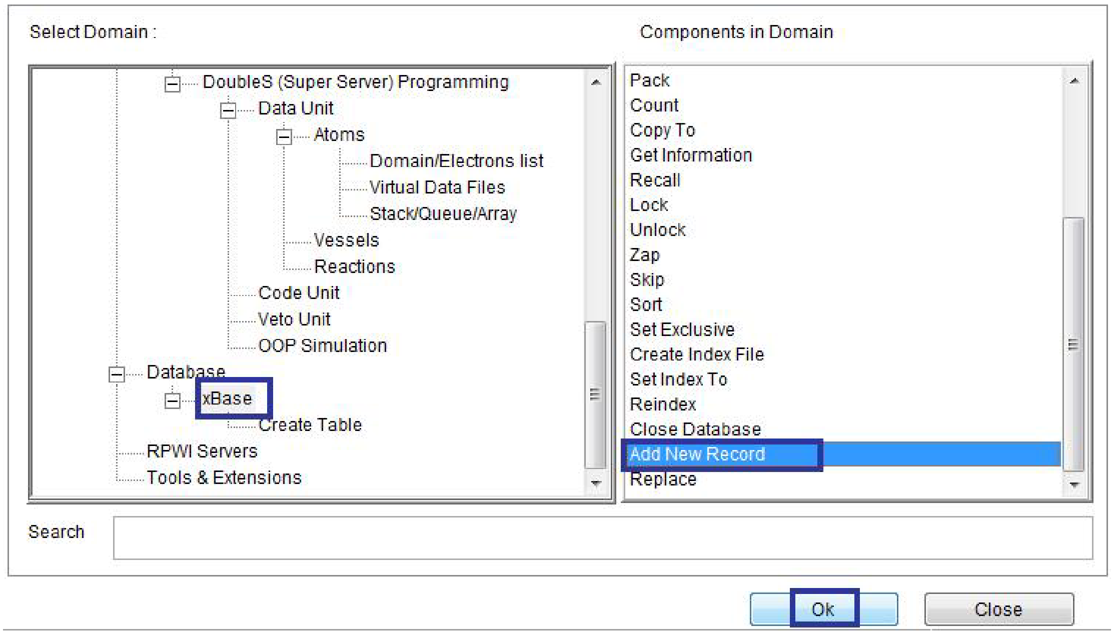

Interaction Page



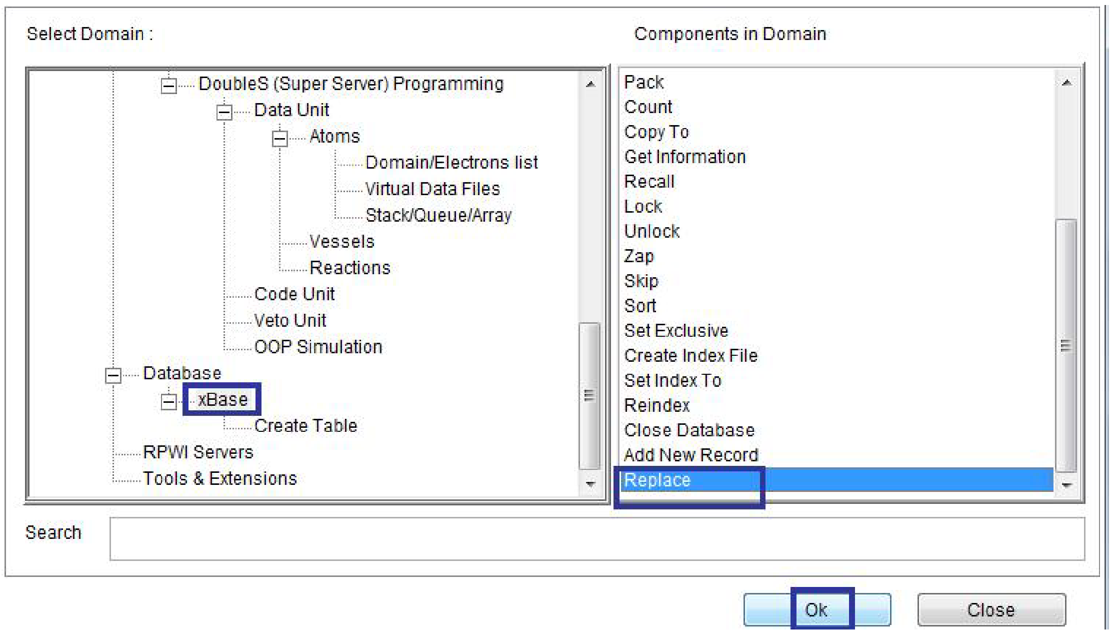

Domain (xBase) Component (Open Table)



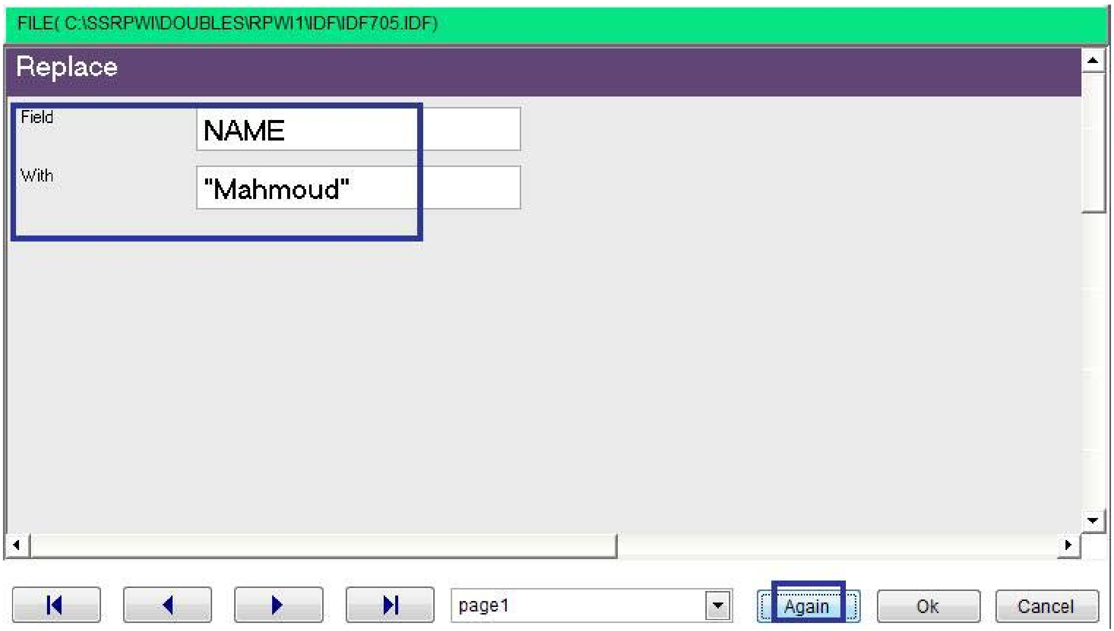

Interaction Page



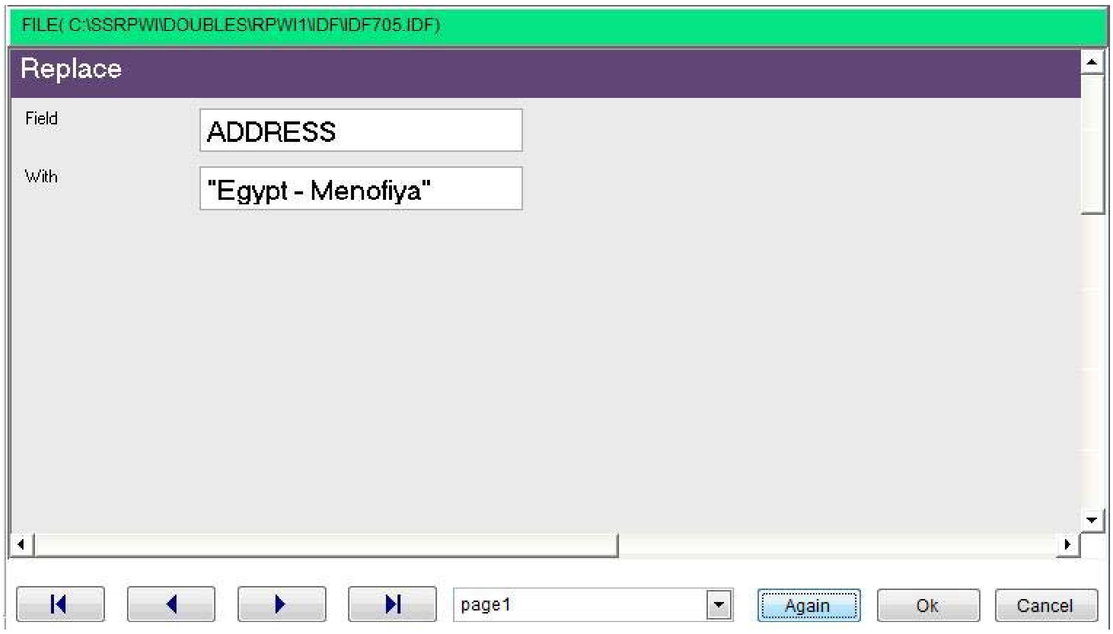

Domain (xBase) Component (Add New Record)



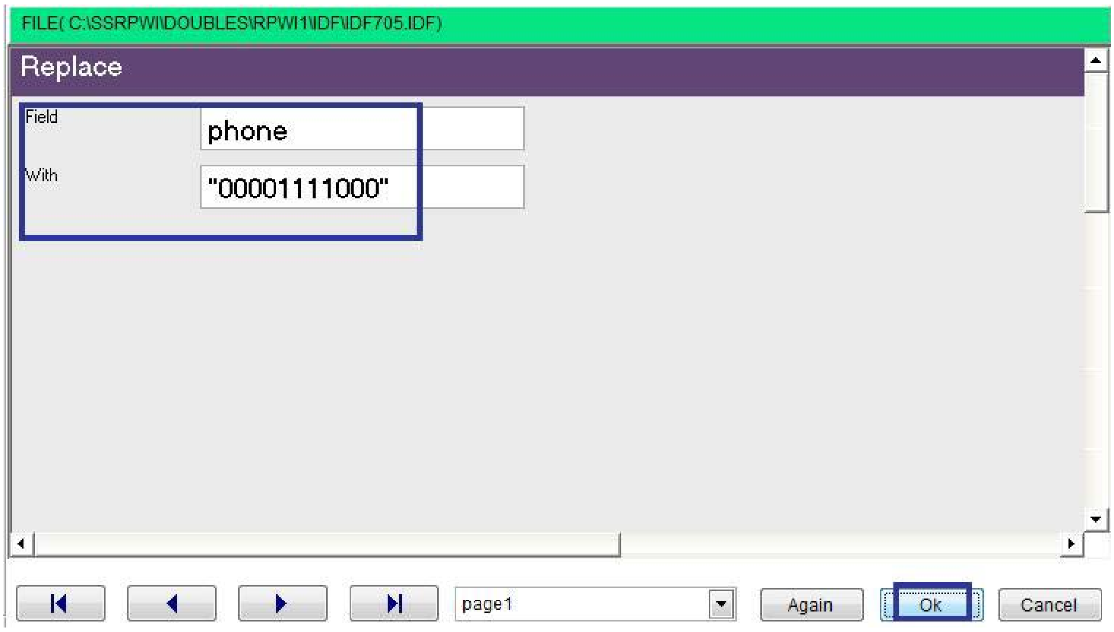

Domain (xBase) Component (Replace)



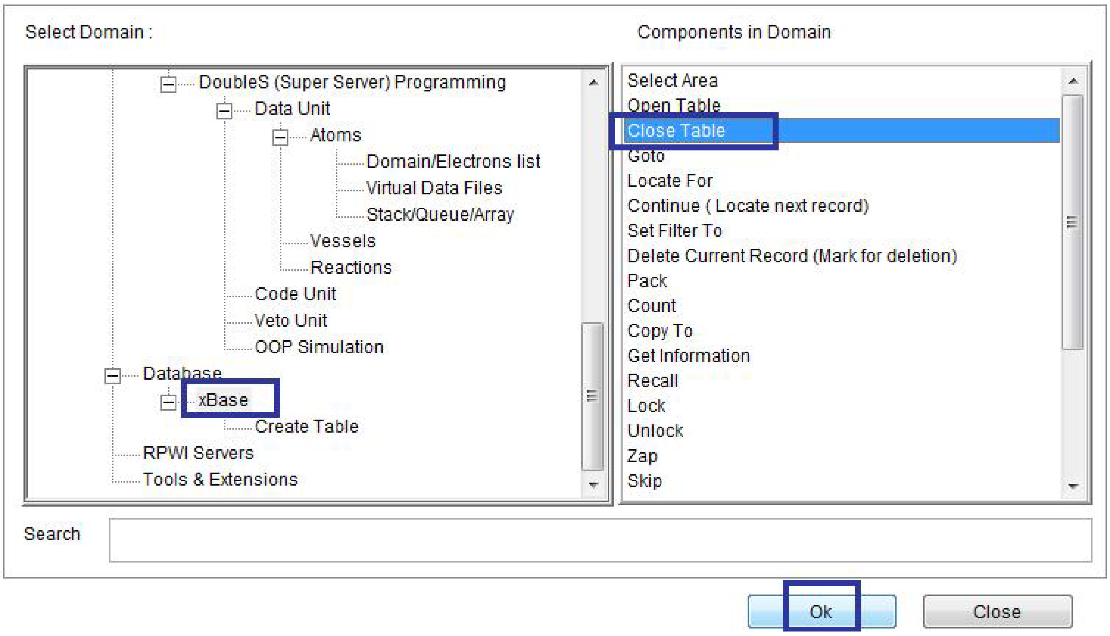

Interaction Page



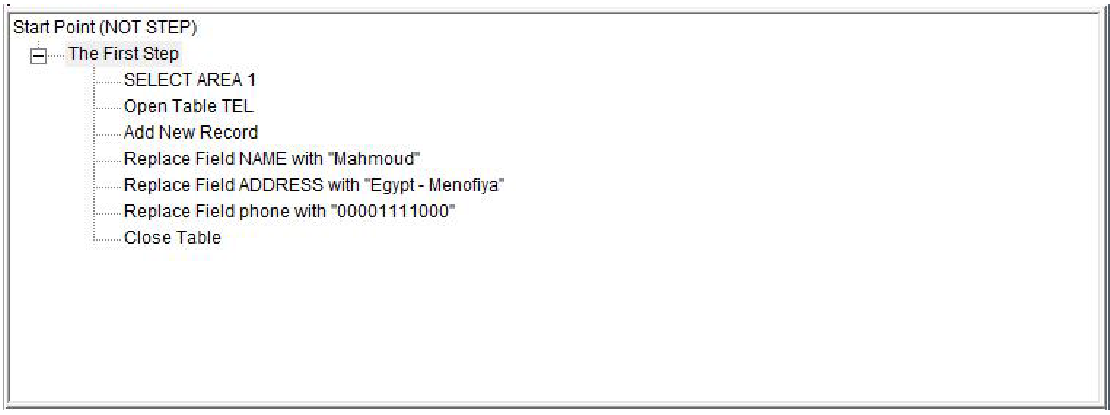

Interaction Page



Interaction Page

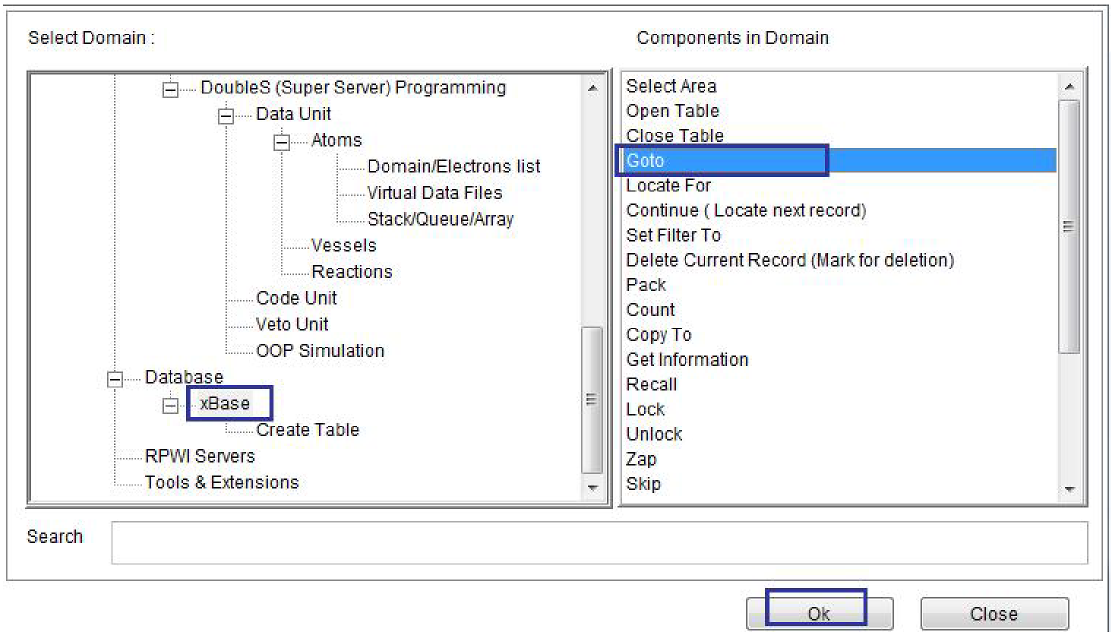


Domain (xBase) Component (Close Table)

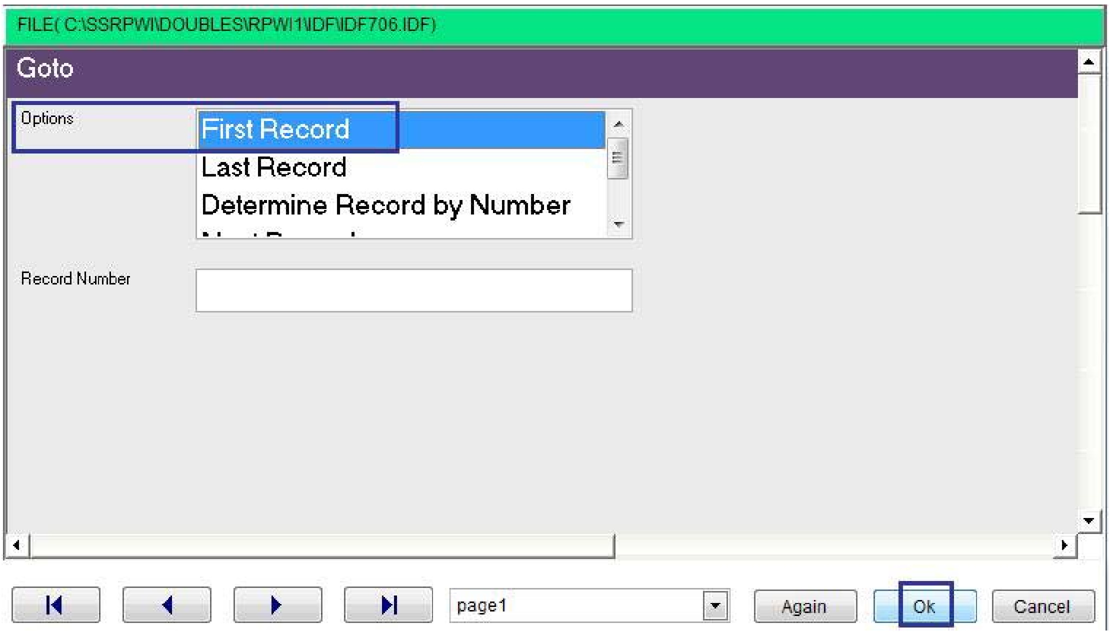


Steps Tree

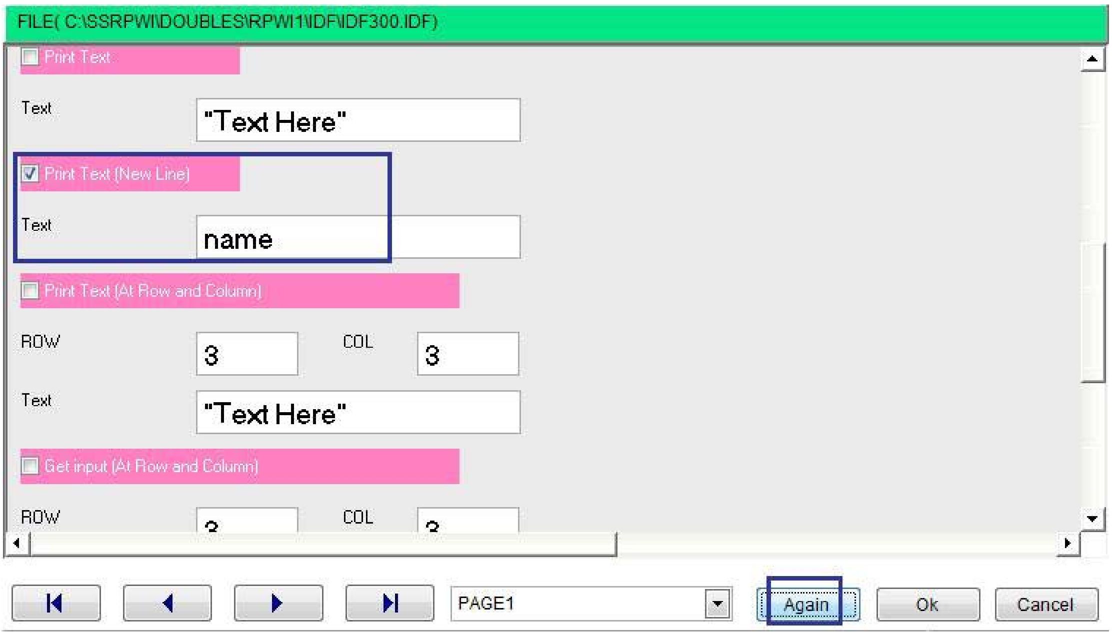
# Moving between records



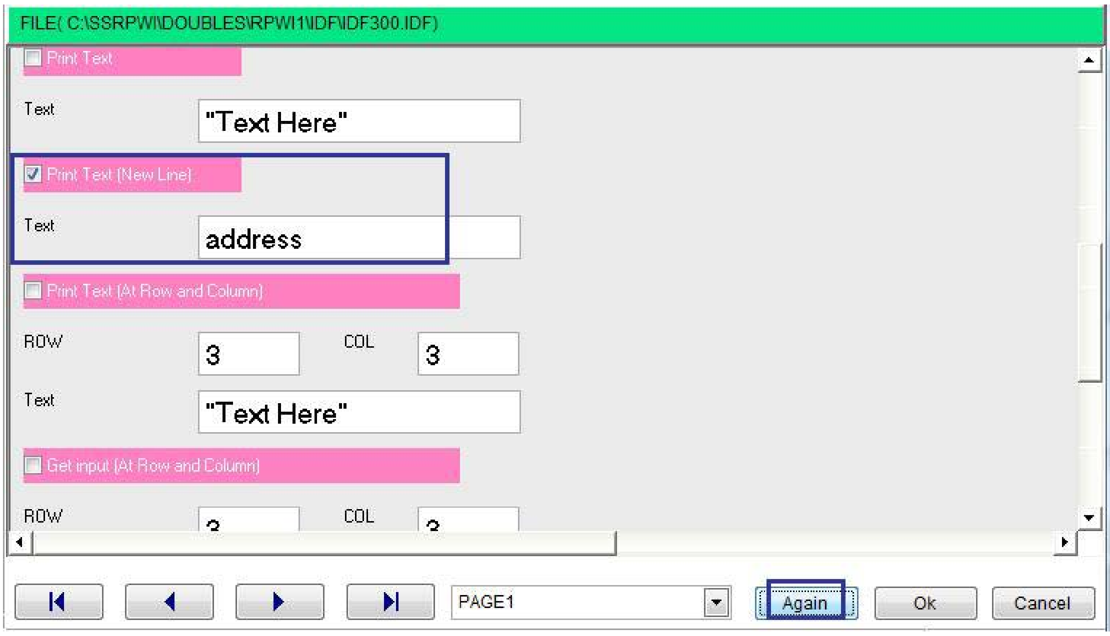

Domain (xBase) Component (Goto)



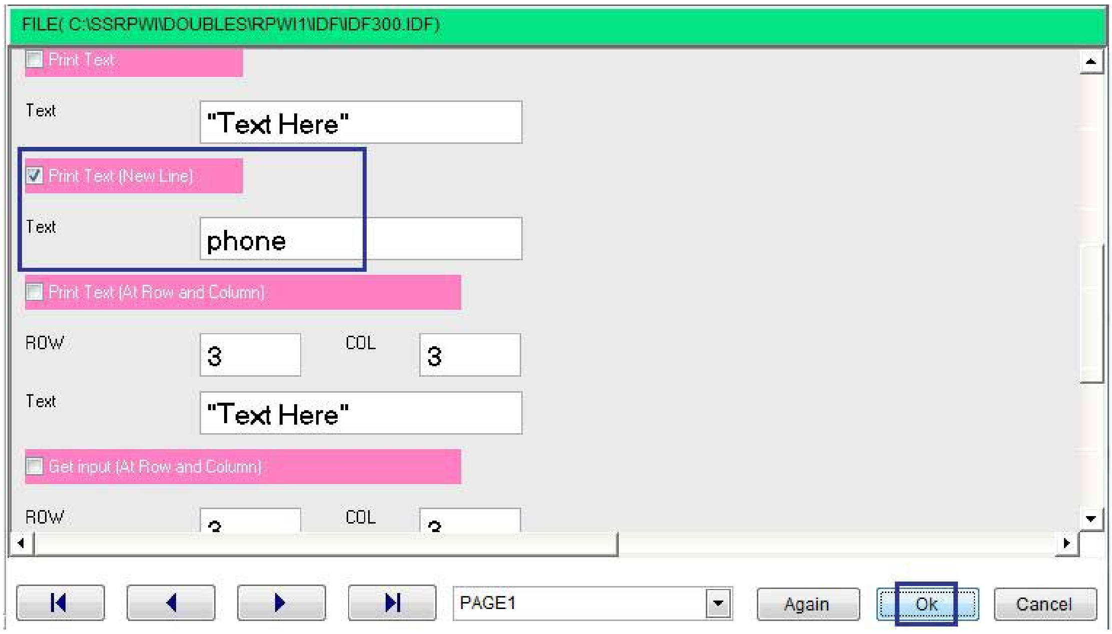

Interaction Page



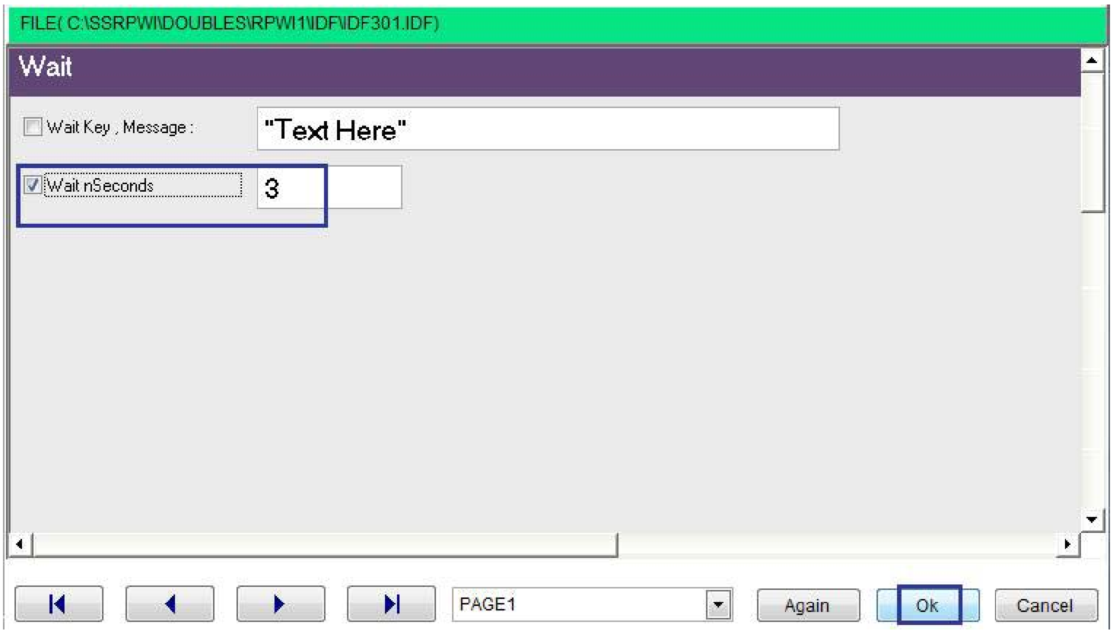

Interaction Page



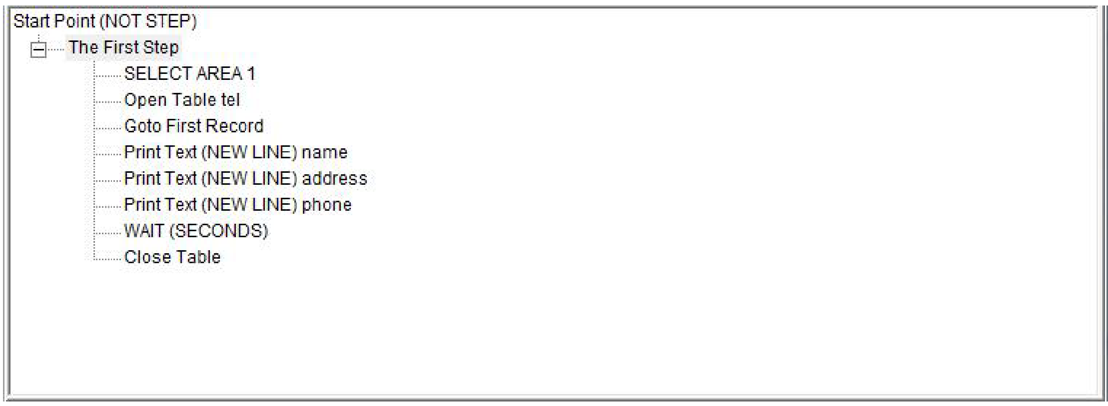

Interaction Page



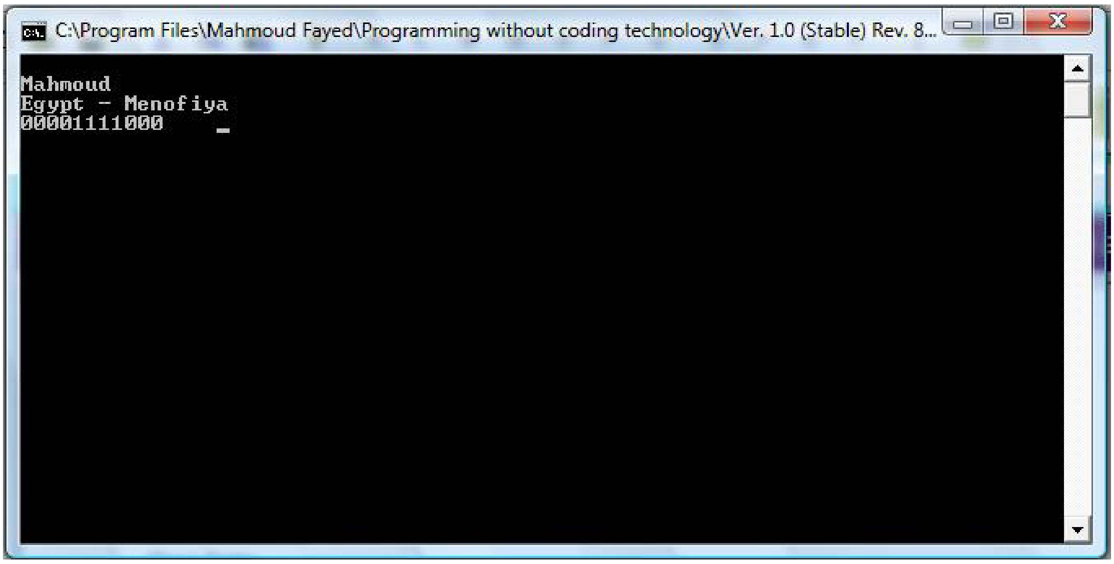

Interaction Page




Interaction Page

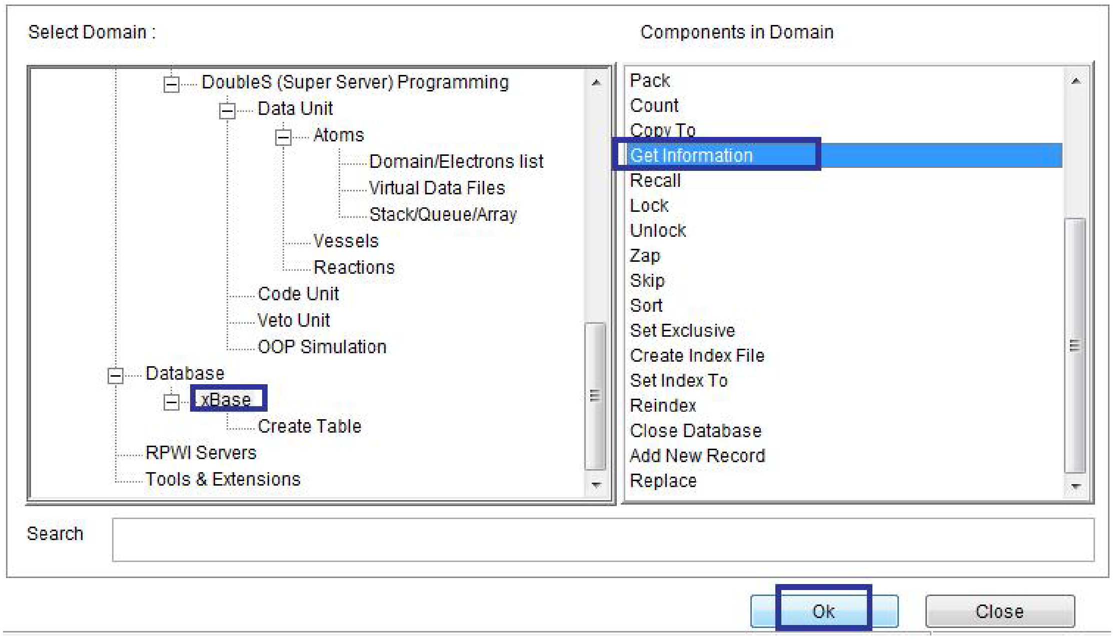



Steps Tree

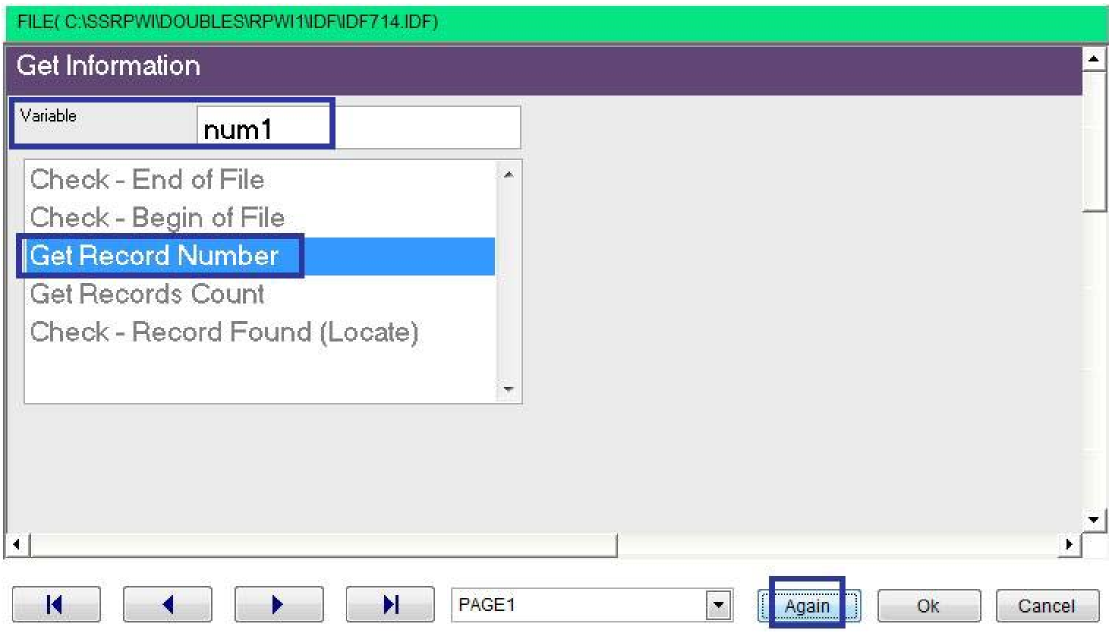



The Final Application

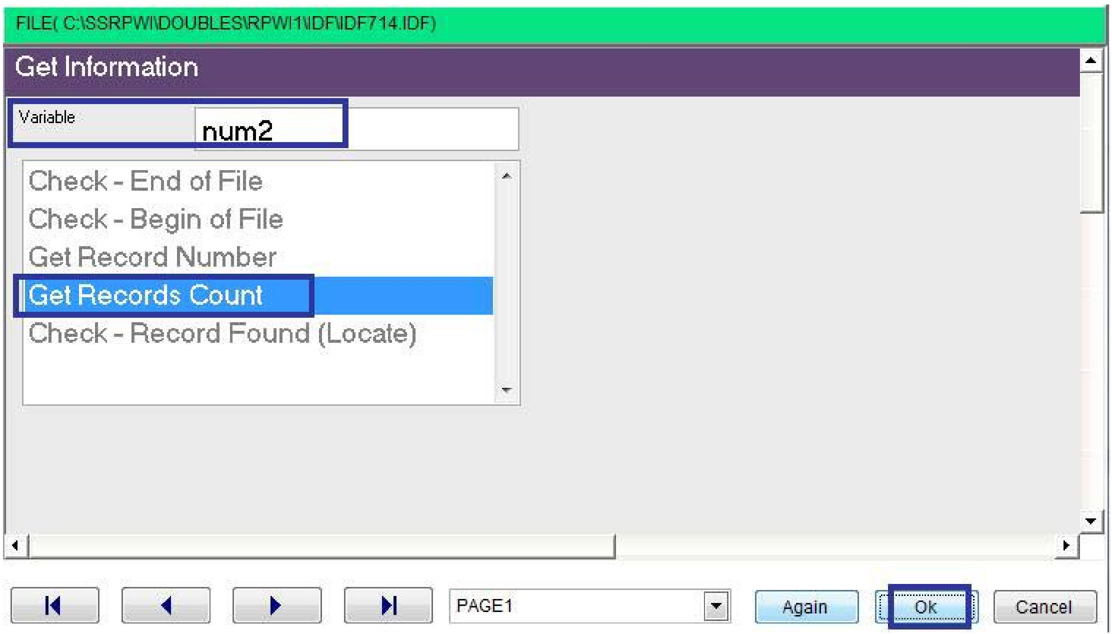
# Record number & records count



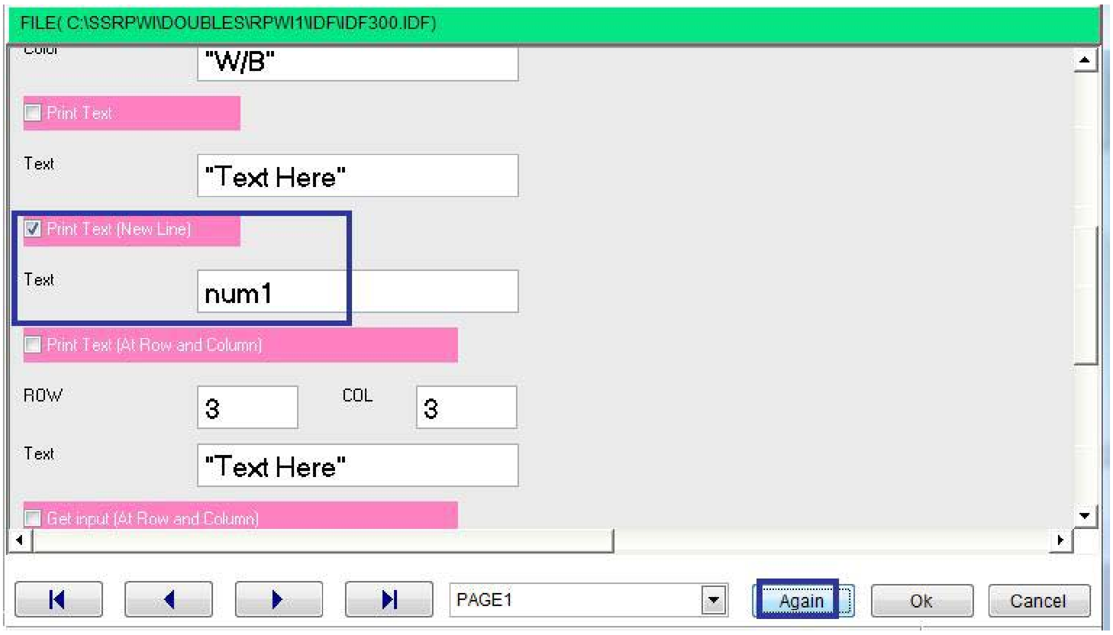

Domain (xBase) Component (Get Information)



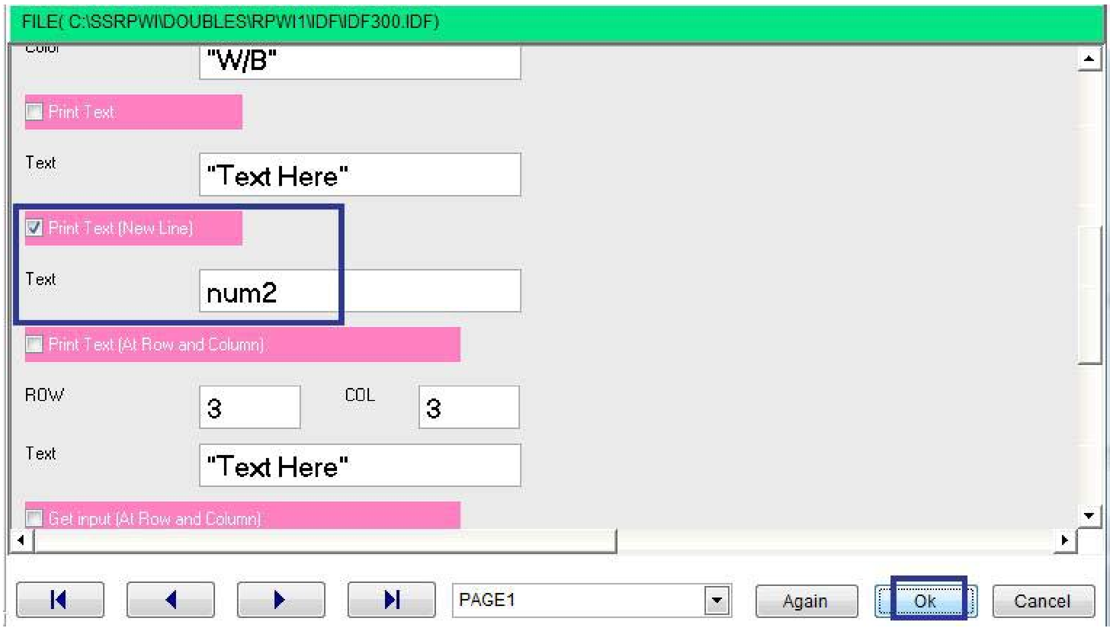

Interaction Page



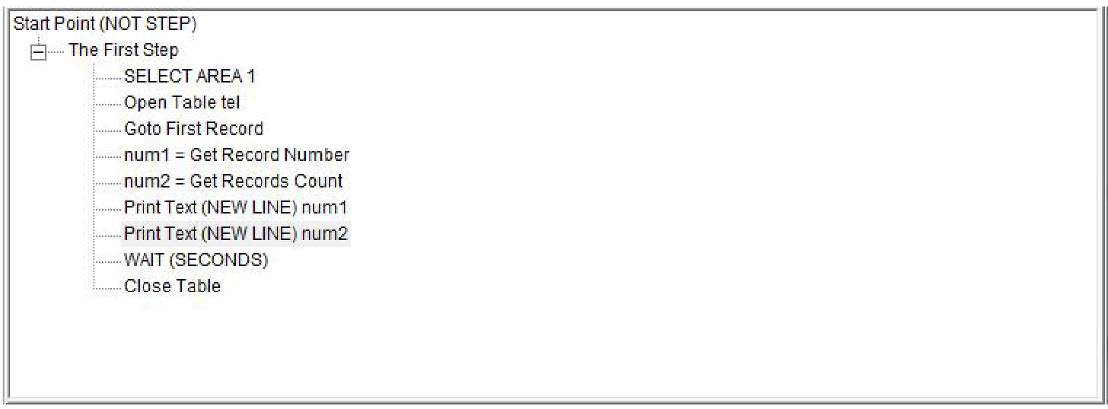

Interaction Page



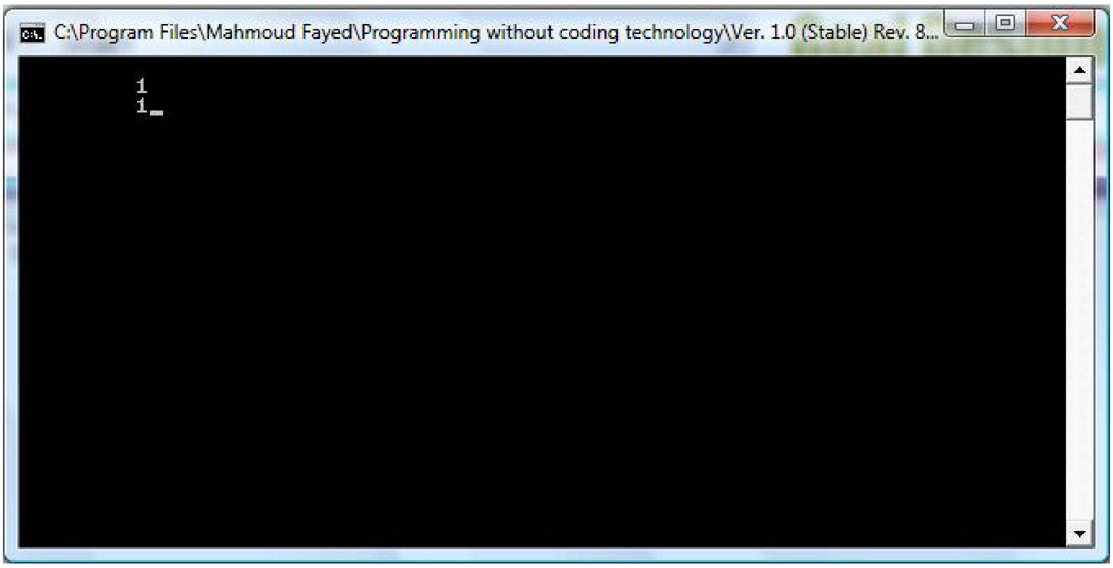

Interaction Page




Interaction Page




Final Steps Tree



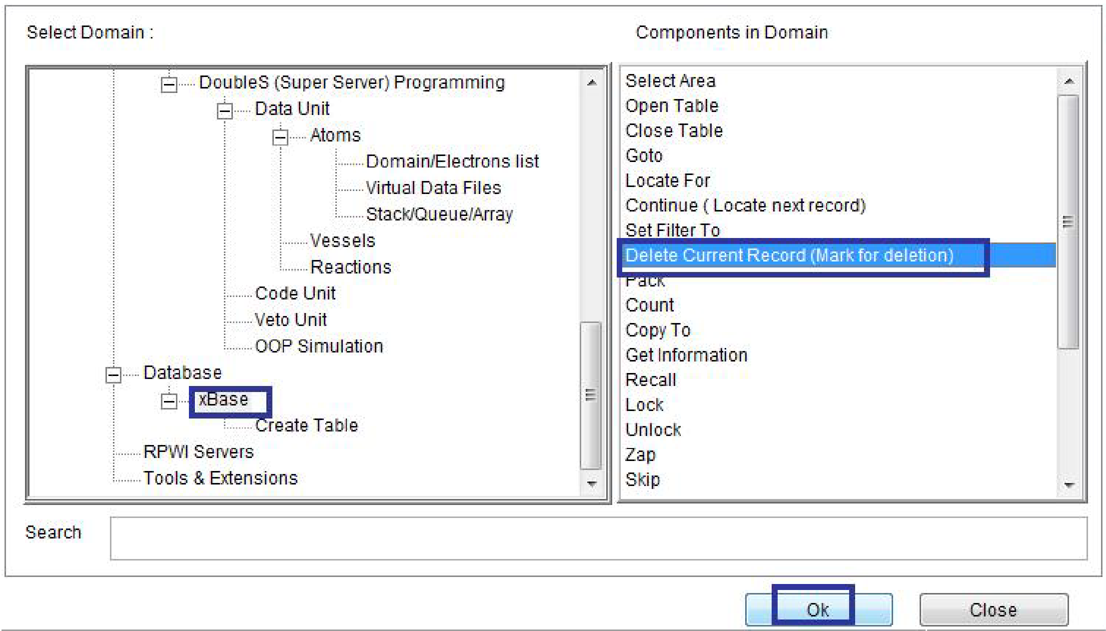
The Final Application

## Modify record

Be sure that this record is the active record, and then you can alter it using
Domain (xBase)
Component (Replace)

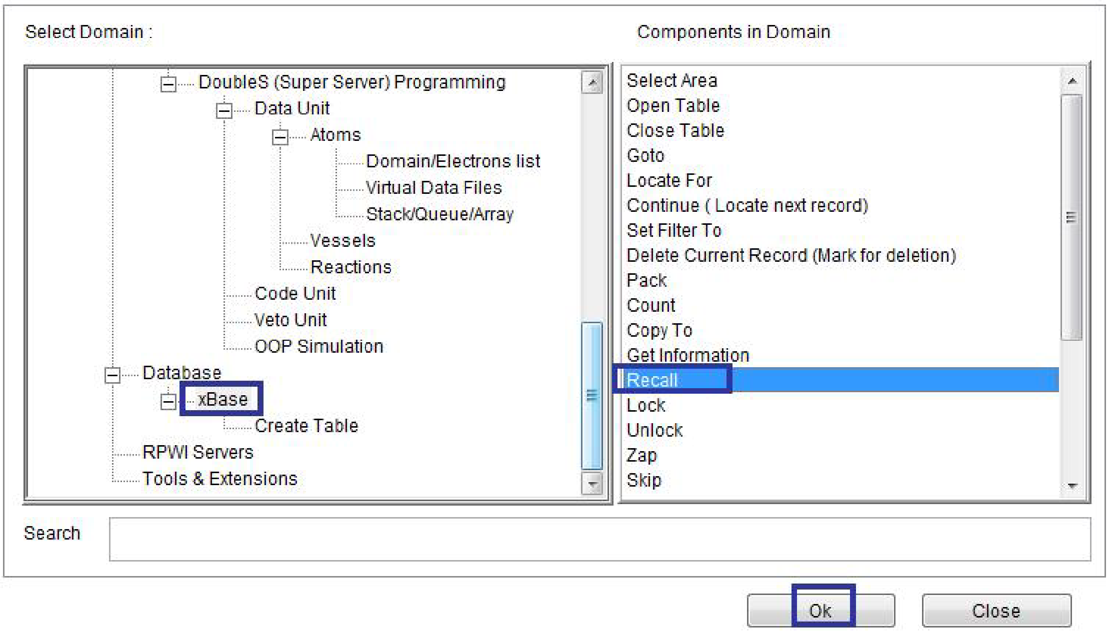
## Deleting records

Be sure that this record is the active record, and then you can mark it for deletion using
Domain (xBase)
Component (Delete Current Record – mark for deletion)



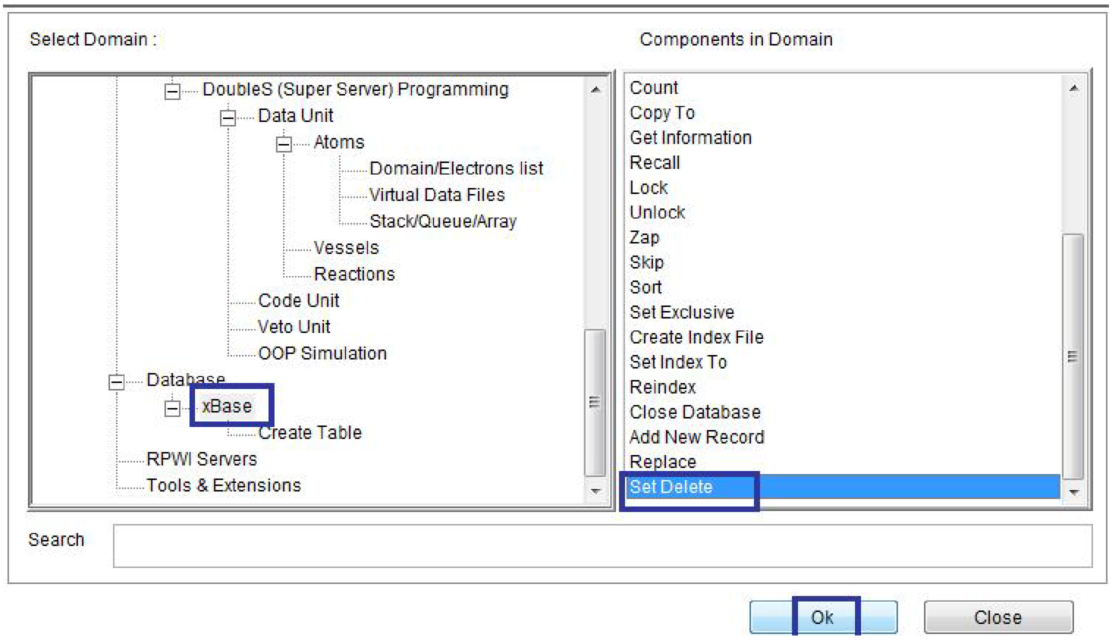
Domain (xBase) Component (Delete Current Record – mark for deletion)

You can recall the deleted record using
Domain (xBase)
Component (Recall)

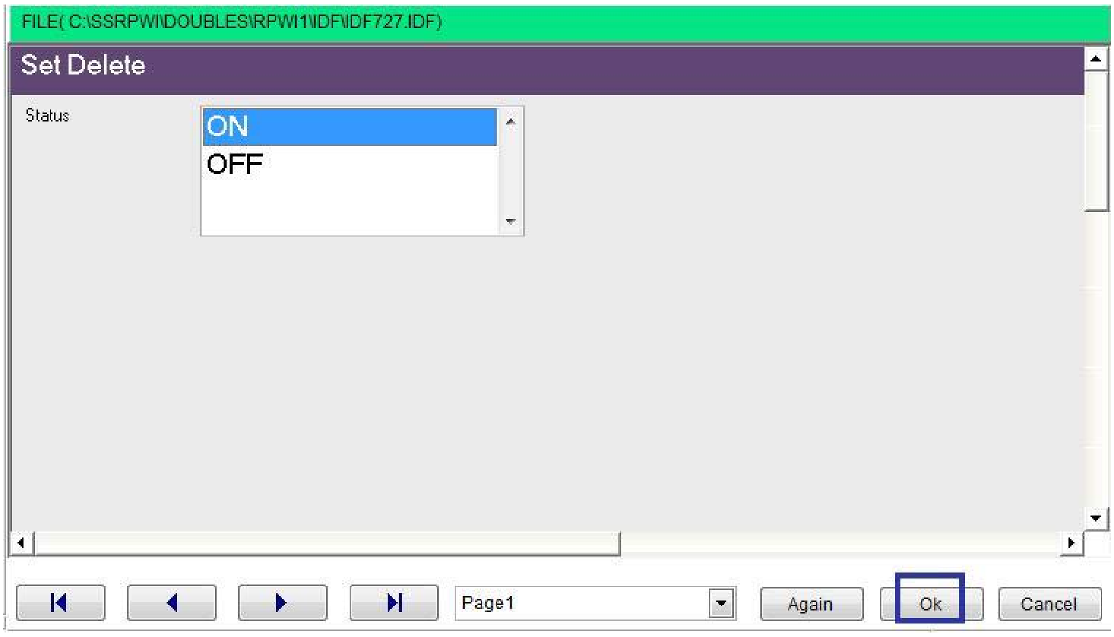



Domain (xBase) Component (Recall)

To exclude the deleted records, you need to set delete on
And to include the deleted records, you need to set delete off
Domain (xBase)
Component (Set Delete)



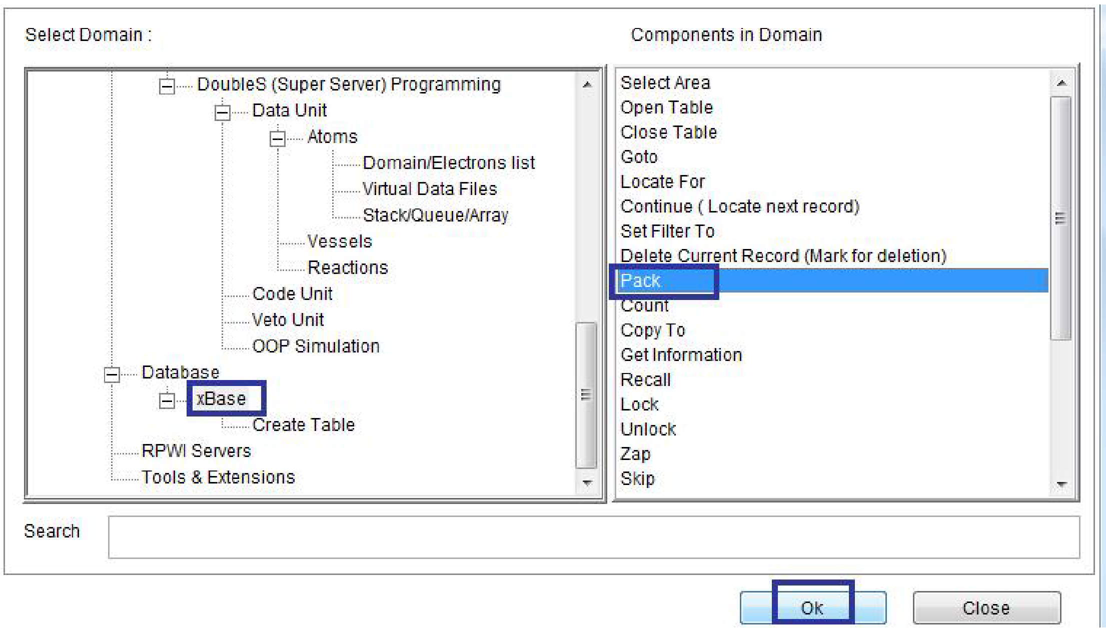

Domain (xBase) Component (Set Delete)




Interaction Page

To pack database file (deleting records marked for deletion)

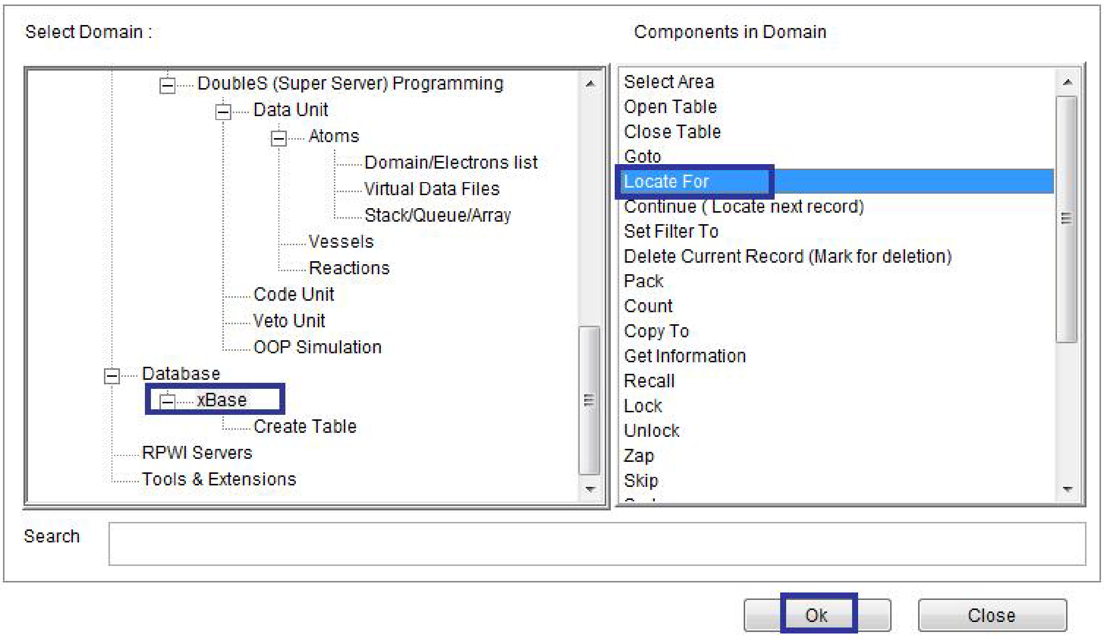
Domain (xBase)
Component (Pack)



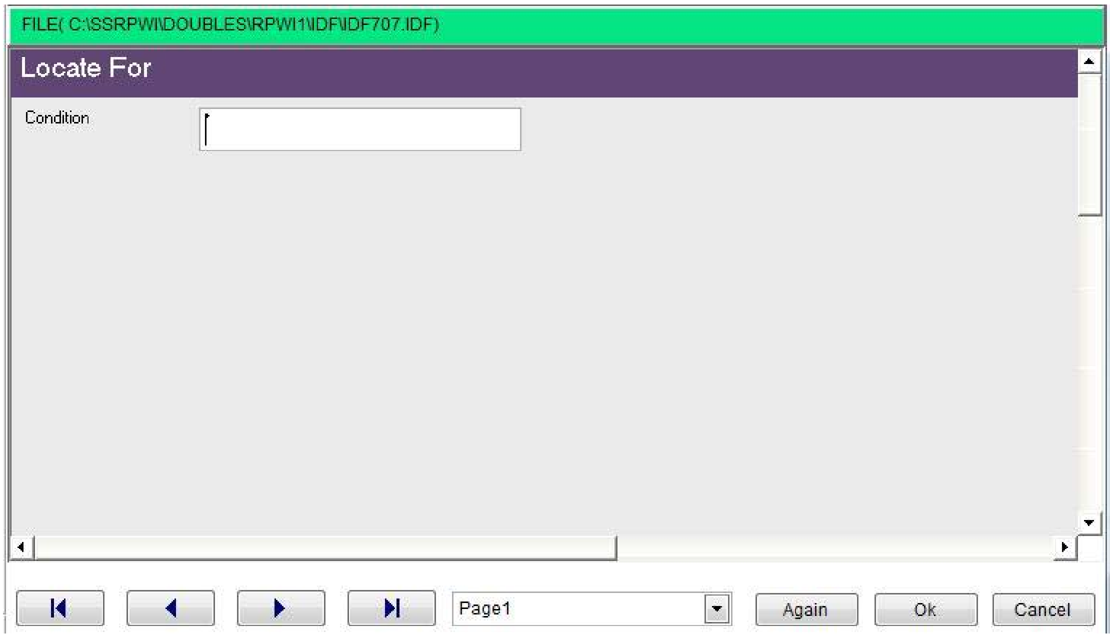

Domain (xBase) Component (Pack)

# Search

You can locate record which match a condition

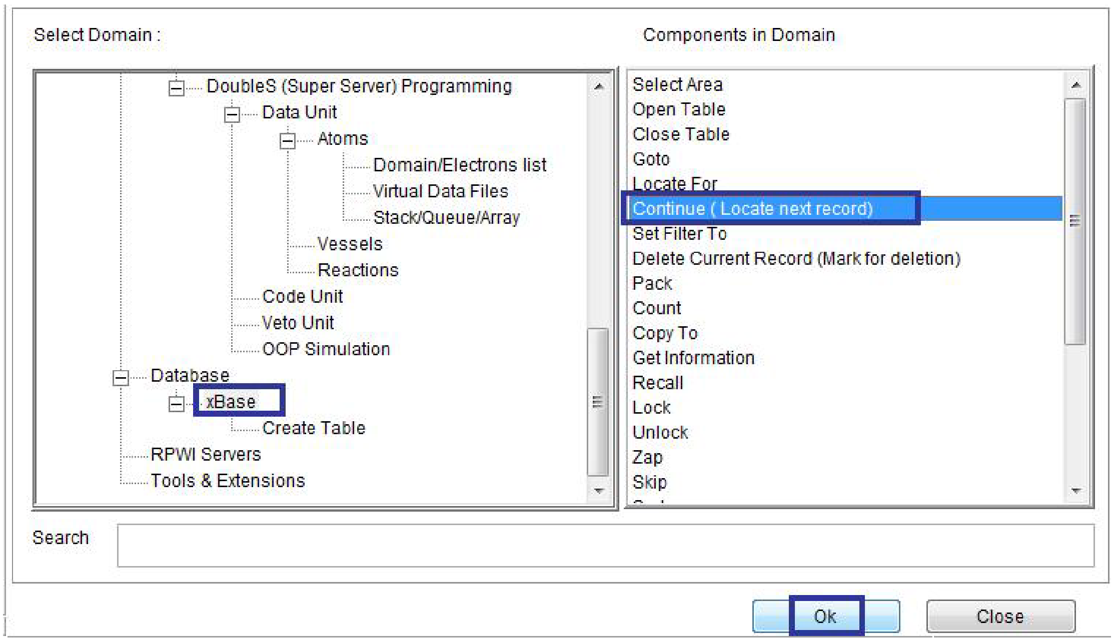



Domain(xBase) Component (Locate For)



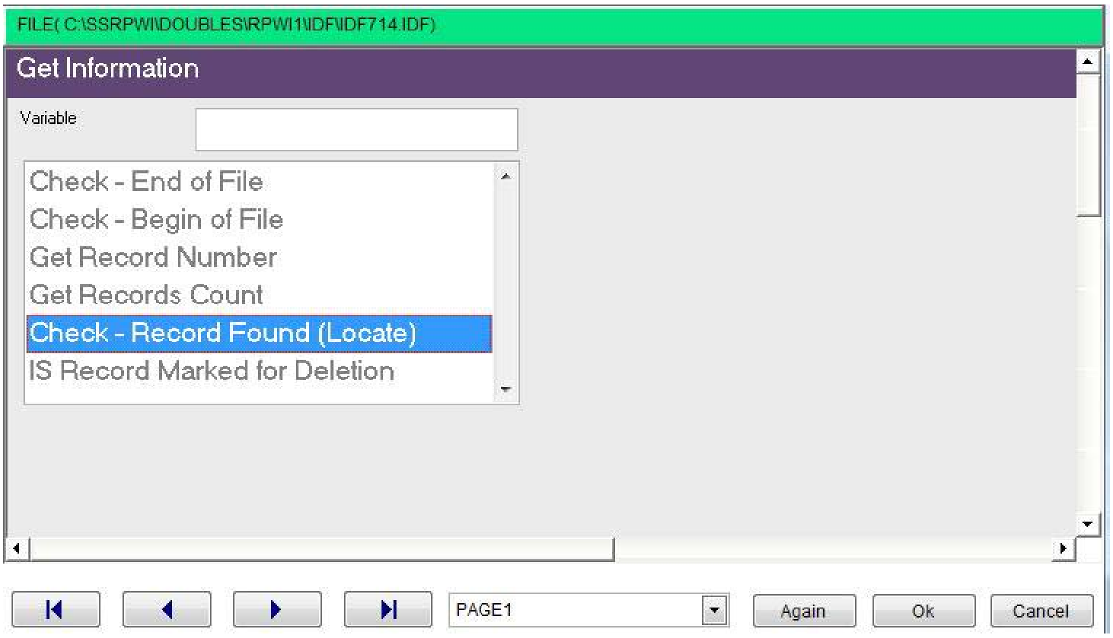

Interaction Page

Also you can continue search to locate another record



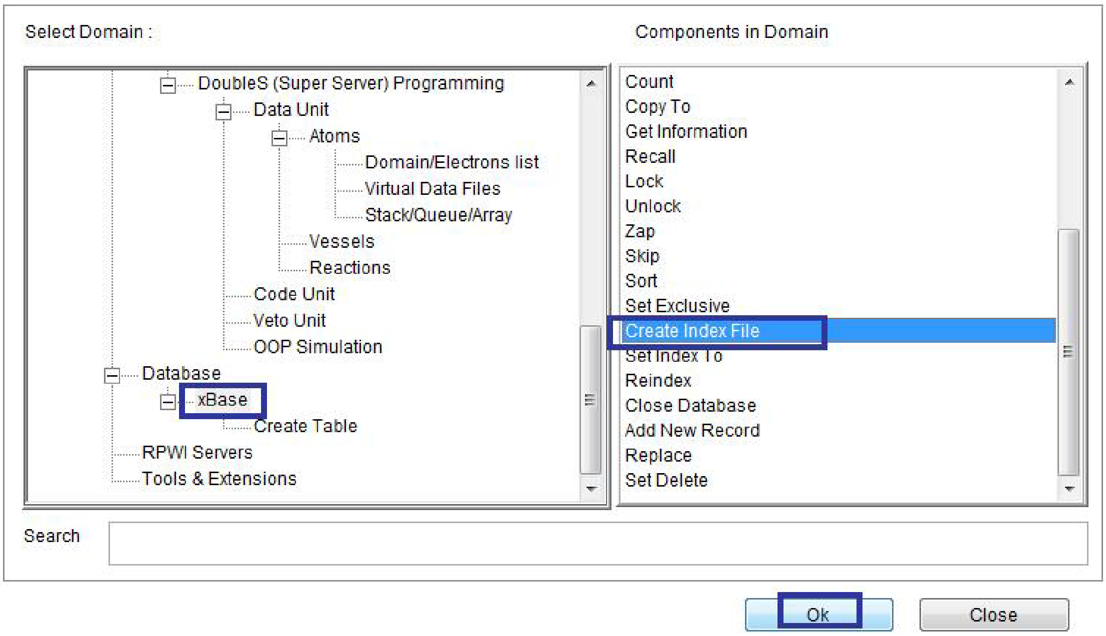
Domain (xBase) Component (Continue)

And after locate of continue you can know if the record is found of not

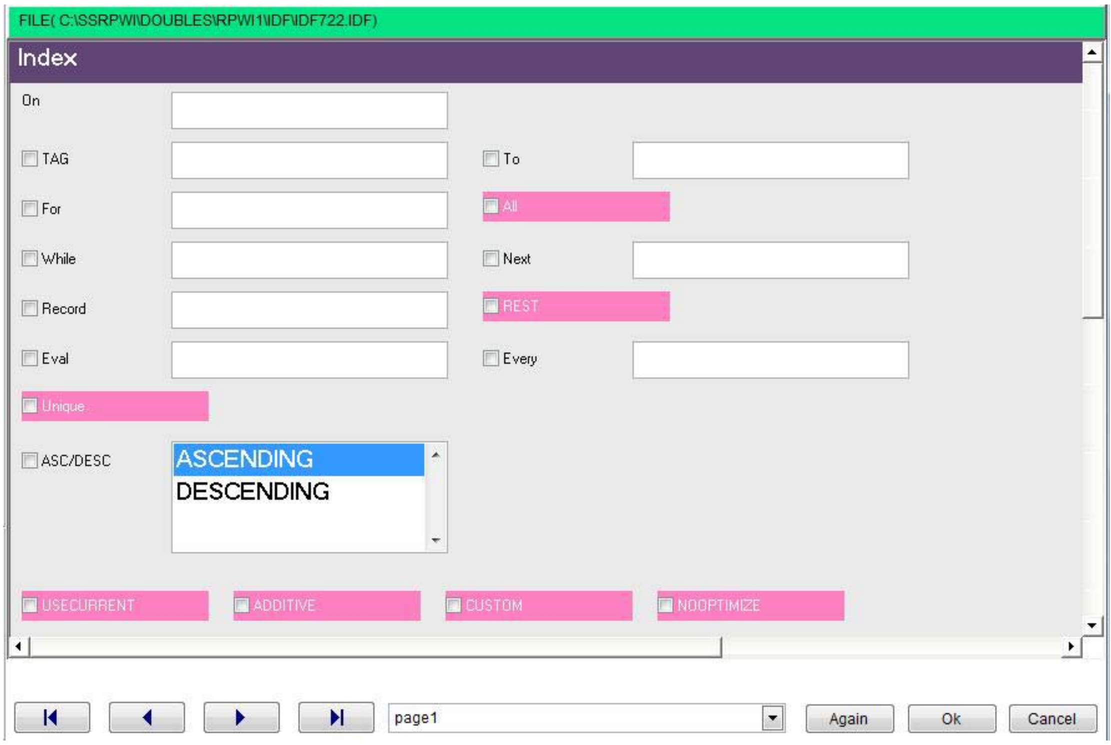


Interaction Page

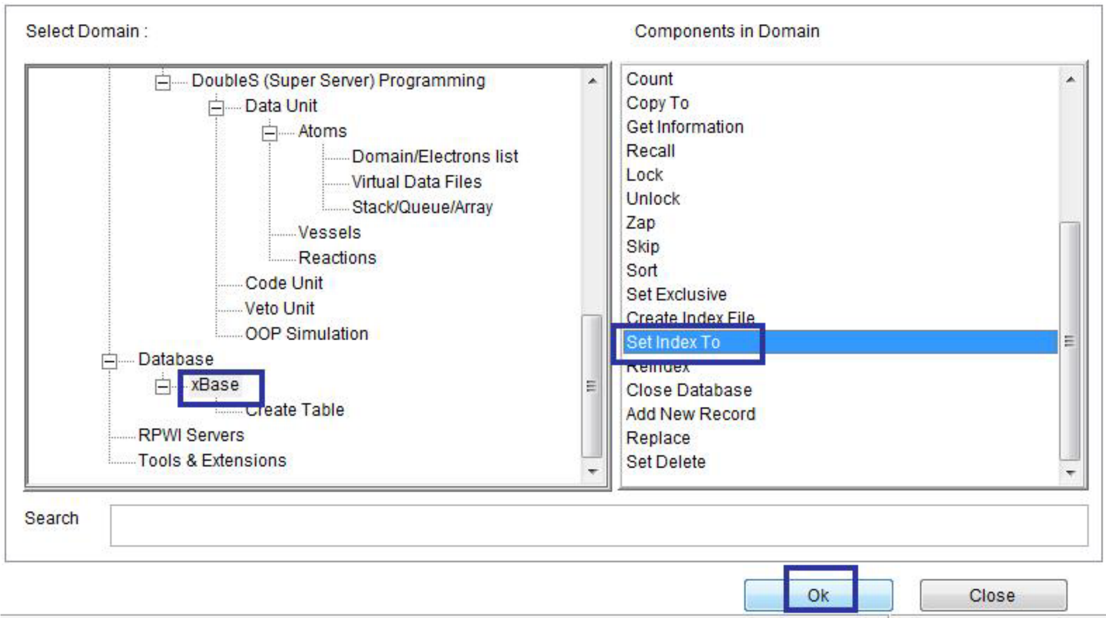
# Index Files



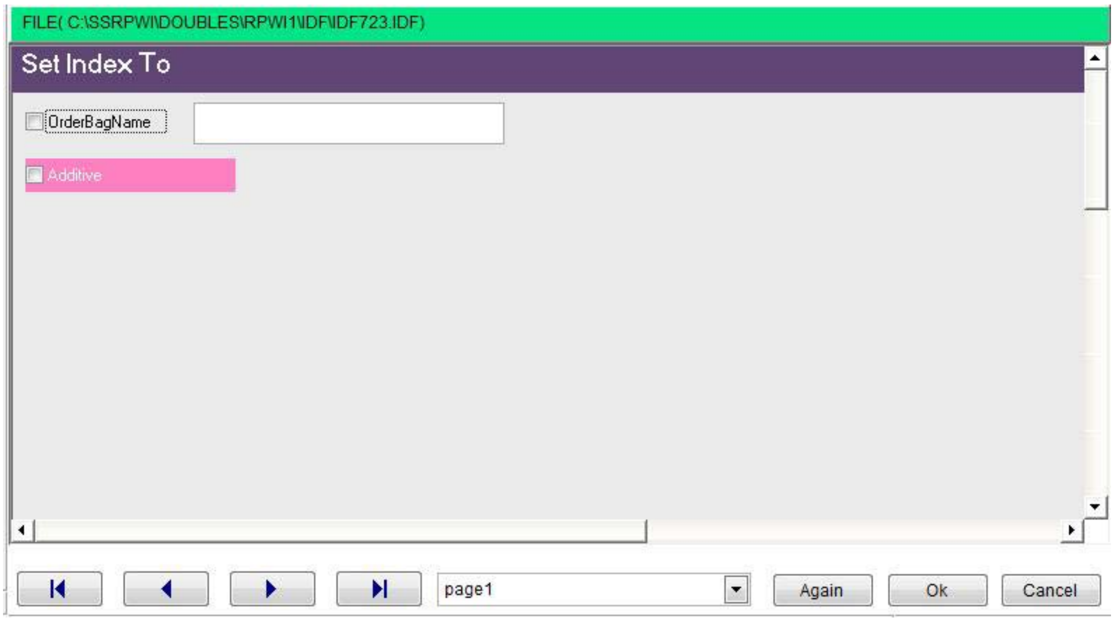

Domain (xBase) Component (Create Index File)



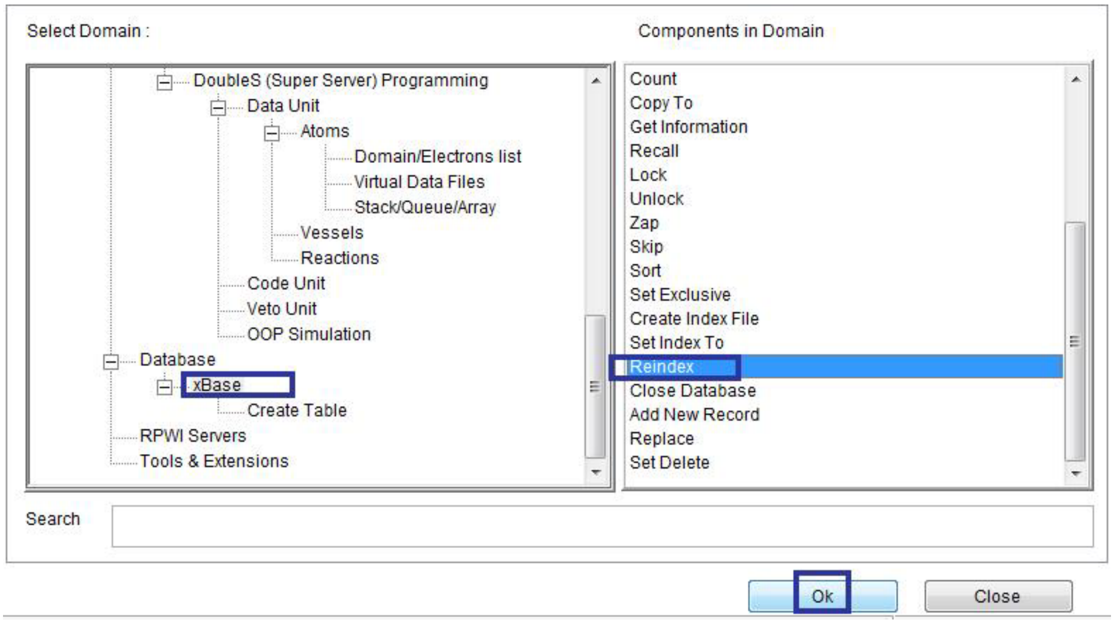

Interaction Page



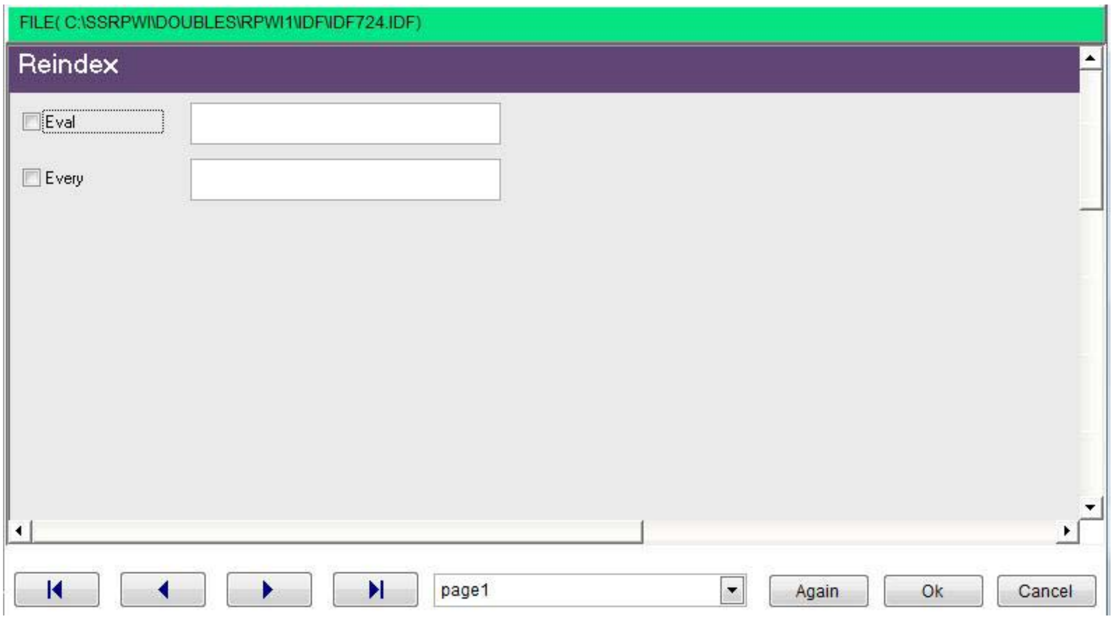

Domain (xBase) Component (Set Index To)




Interaction Page



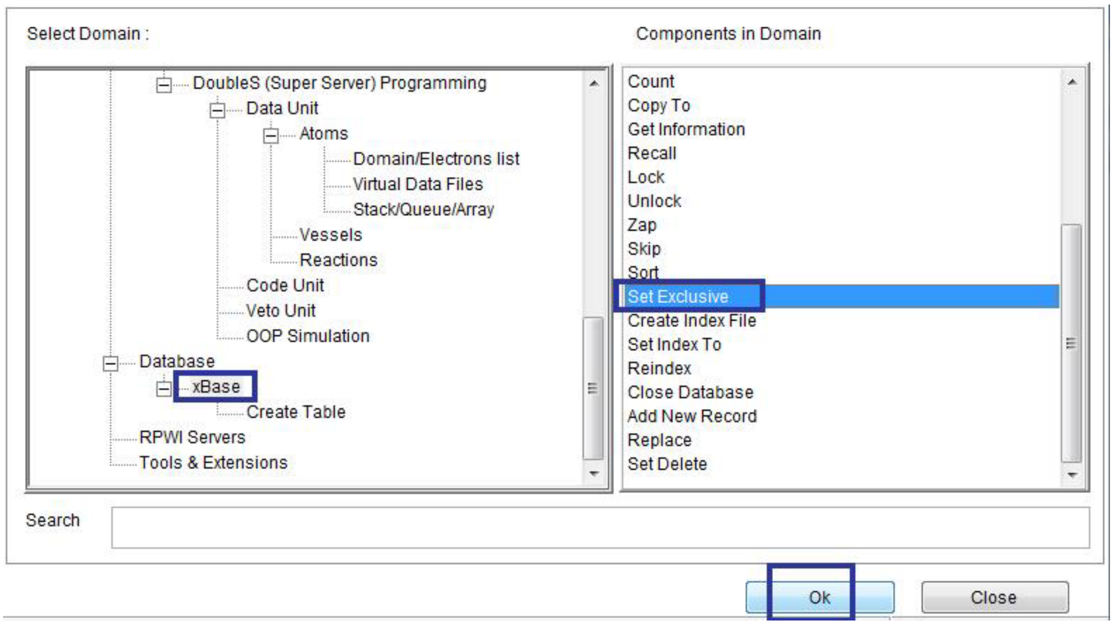

Domain (xBase) Component (Reindex)



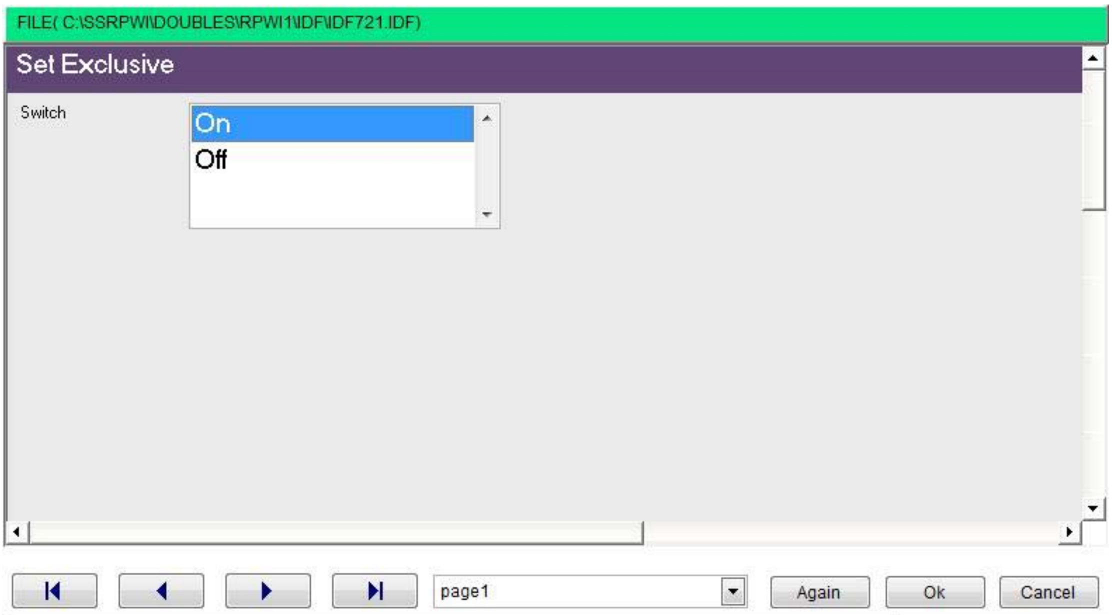

Interaction Page

# Shared Data

To open database files as shared table
You need to set exclusive off before opening the table



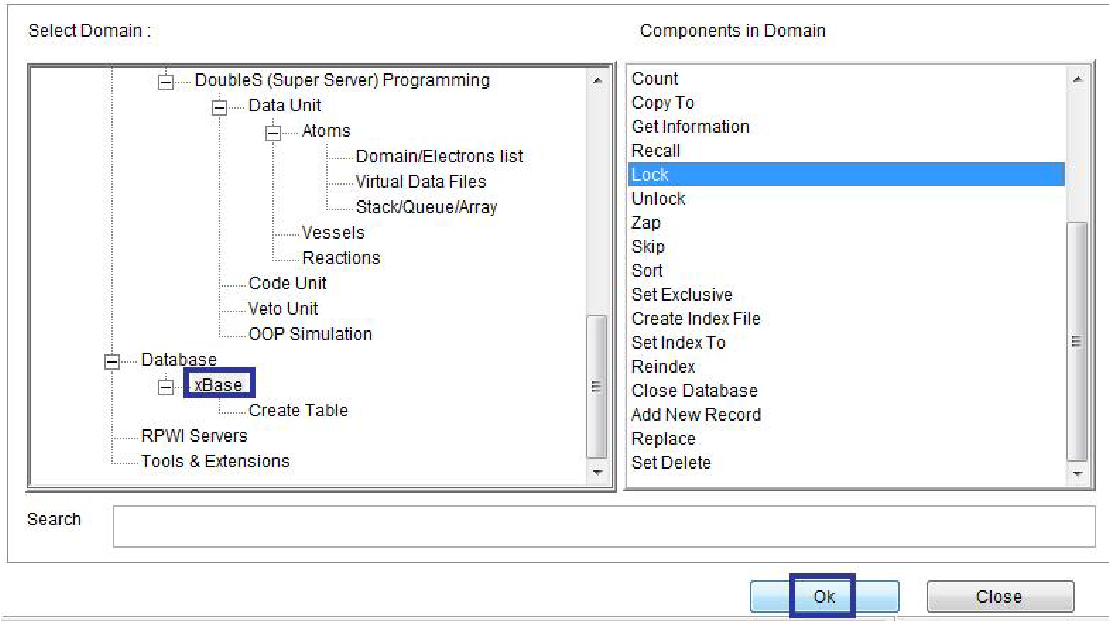

Domain (xBase) Component (Set Exclusive)



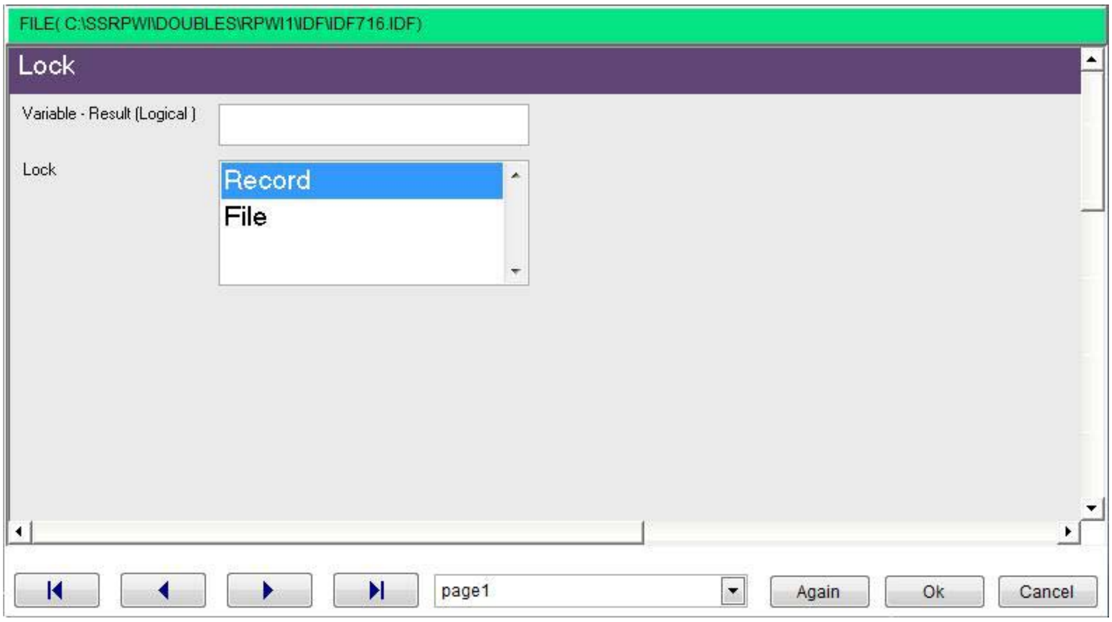

Interaction Page

To modify record data in shared environment you need to lock the record before altering the data and unlocking the record after altering the data

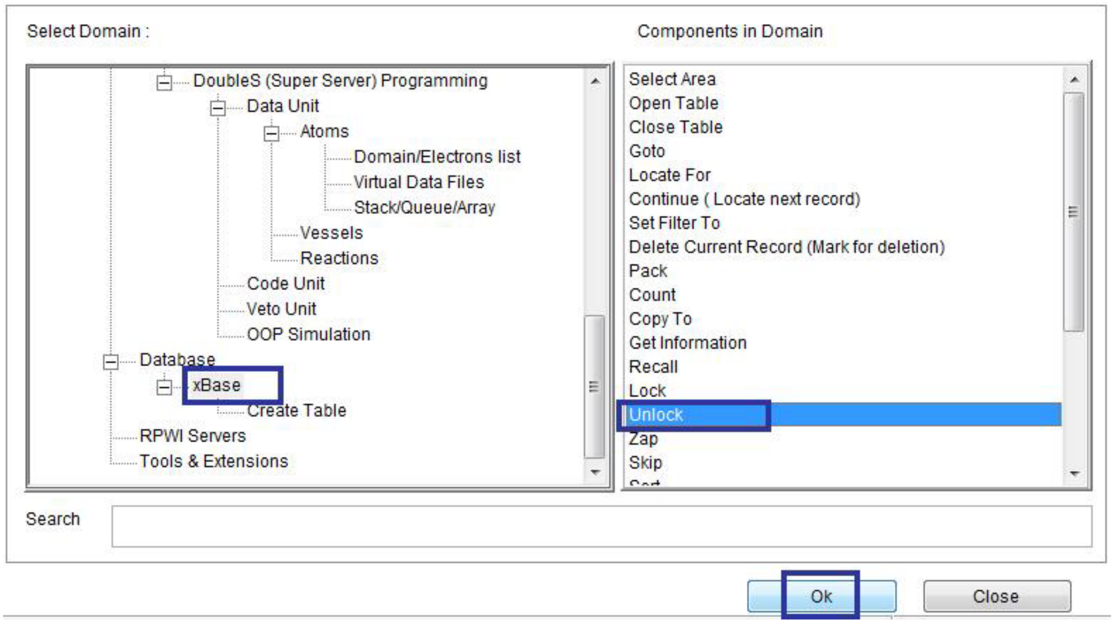
Also you can lock and unlock the data files (all records) before operations that alter all records.




Domain (xBase) Component (Lock)



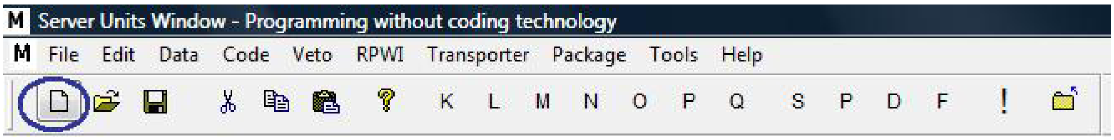

Interaction Page

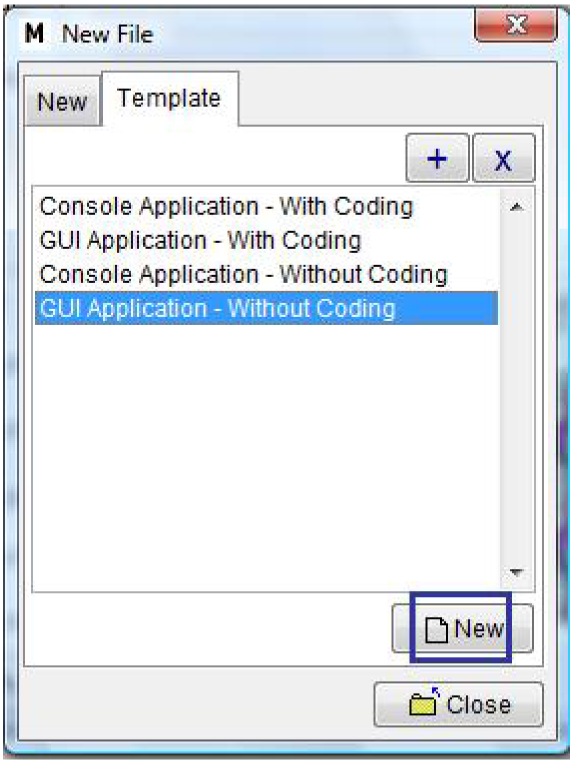


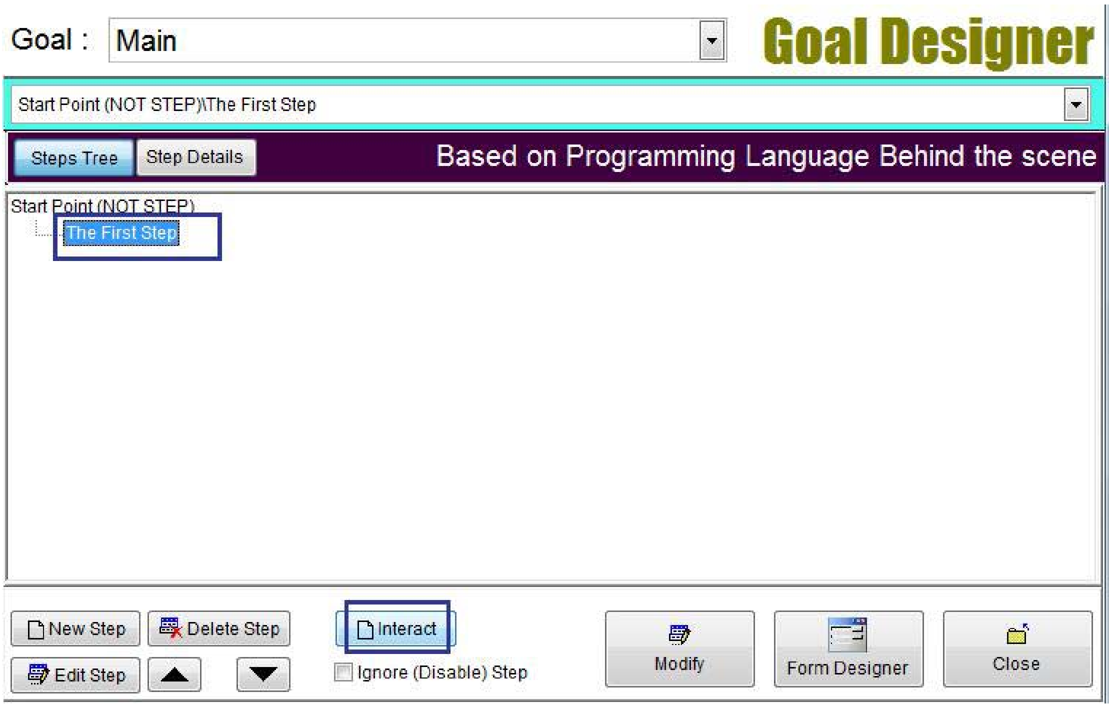

Domain (xBase) Component (Unlock)

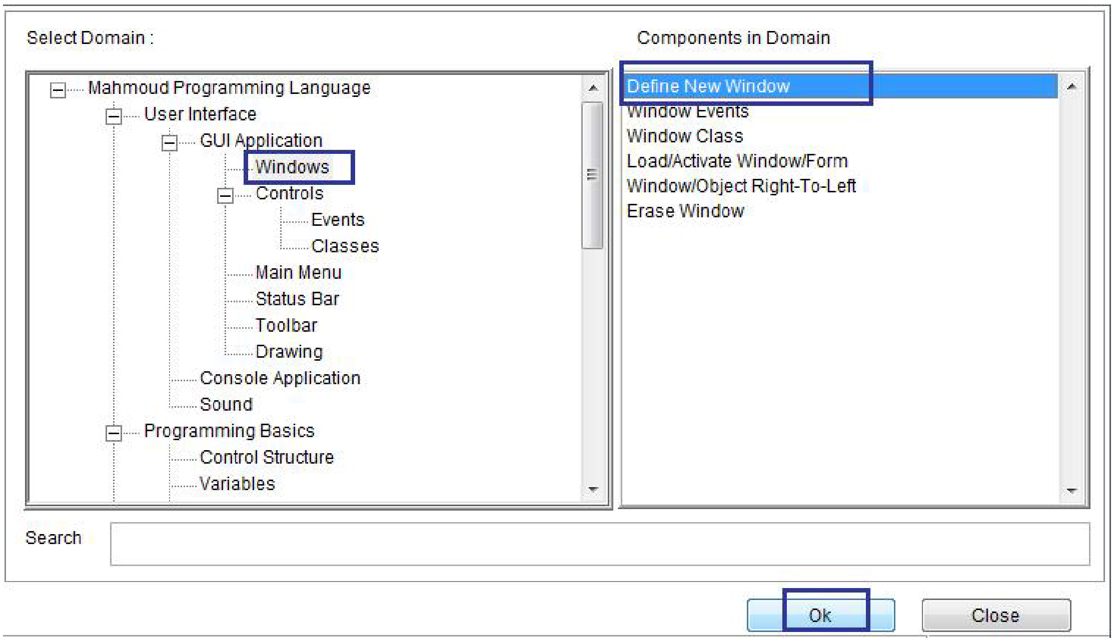
# GUI Applications

We are going to create our first GUI application

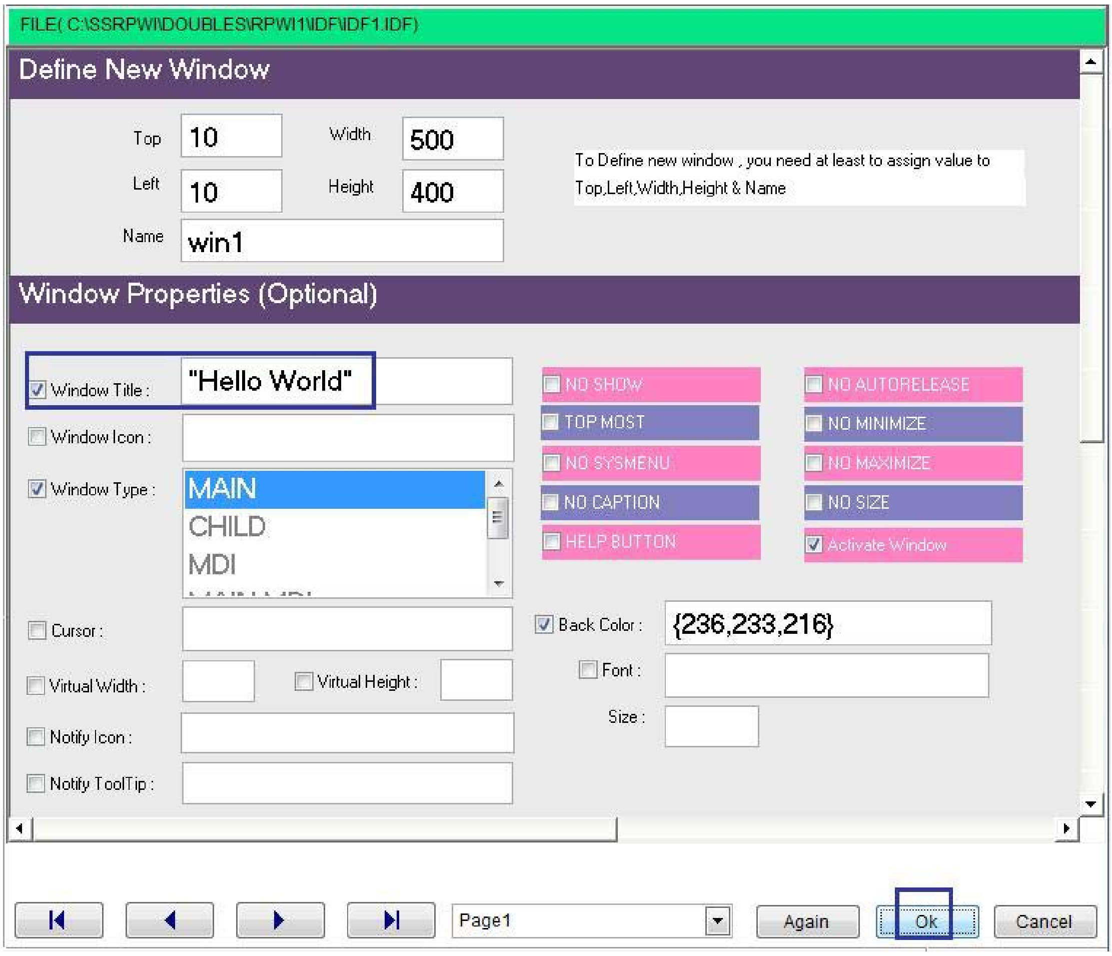




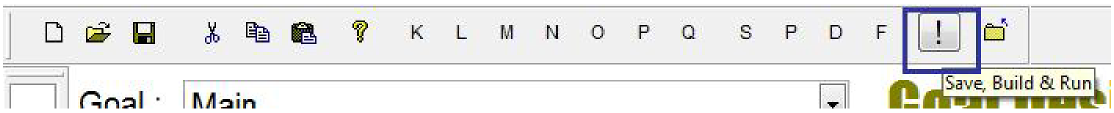

Domain (Windows) Component (Define New Window)

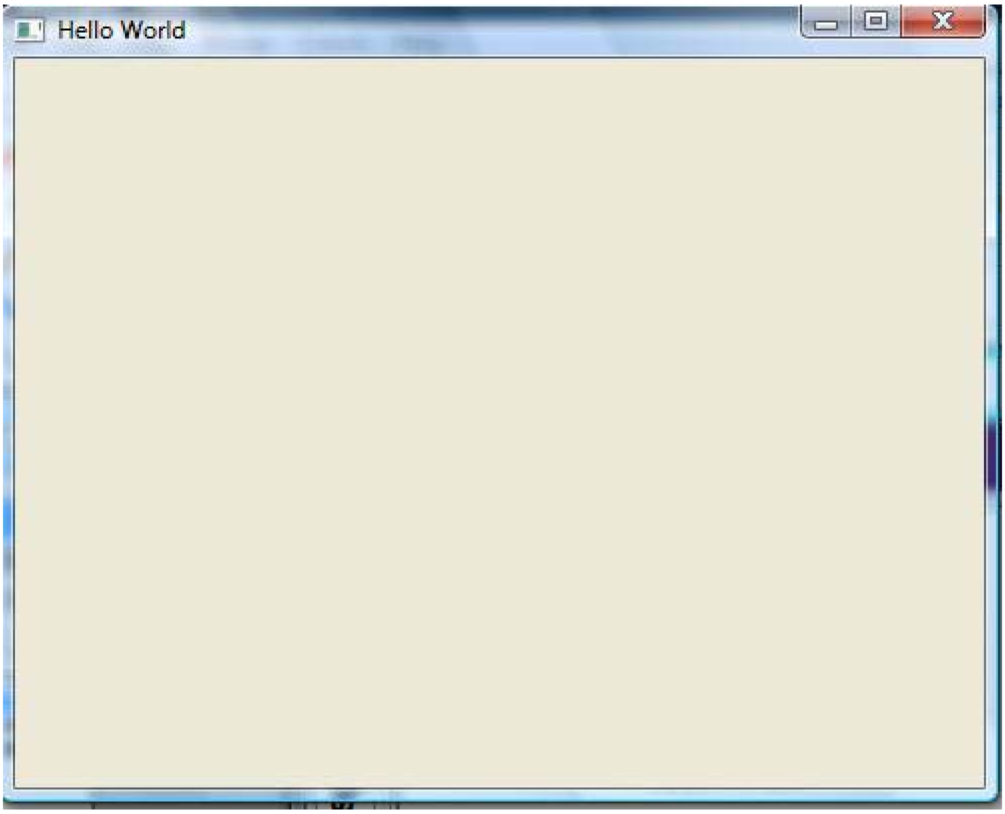



Interaction Page




Steps Tree





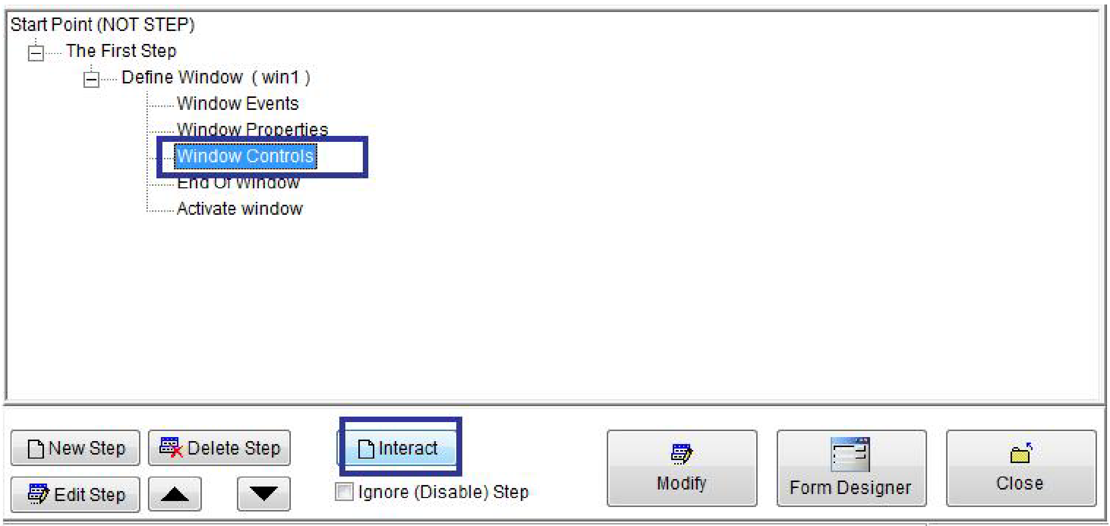
The final application

# GUI – Controls (Objects, Events & Classes)

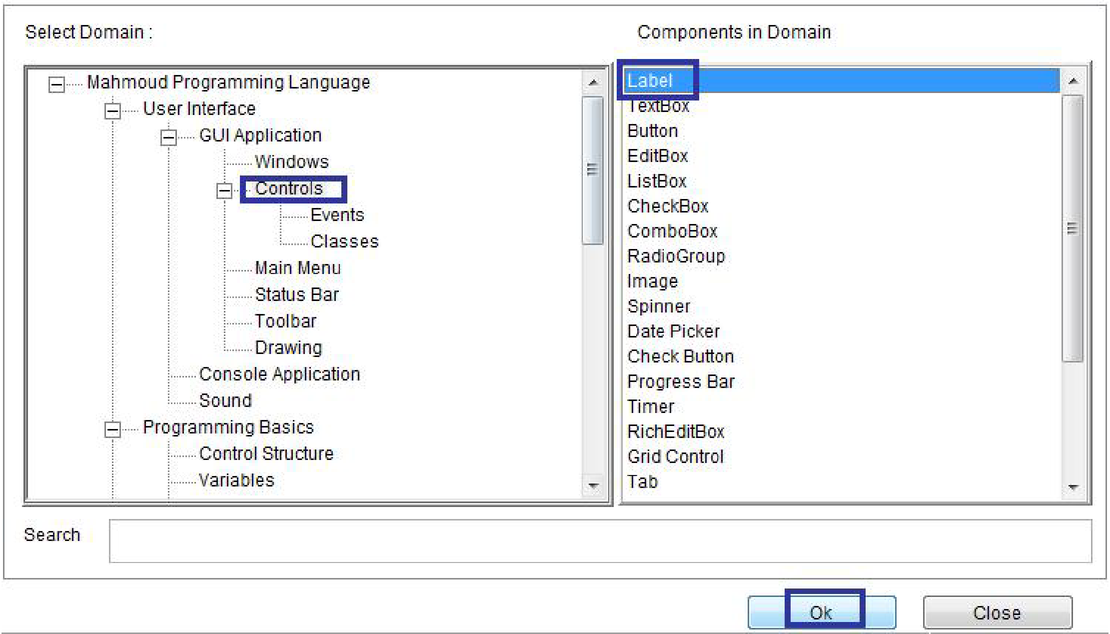
- Label
- TextBox
- Button
- EditBox
- Listbox
- CheckBox
- ComboBox
- RadioGroup
- Image
- Spinner
- DatePicker
- CheckButton
- ProgressBar
- Timer
- RichEditBox
- Grid
- Tab
- Tab Page
- Tree
- Slider
- Frame

- HyperLink
- AnimateBox
- Browse

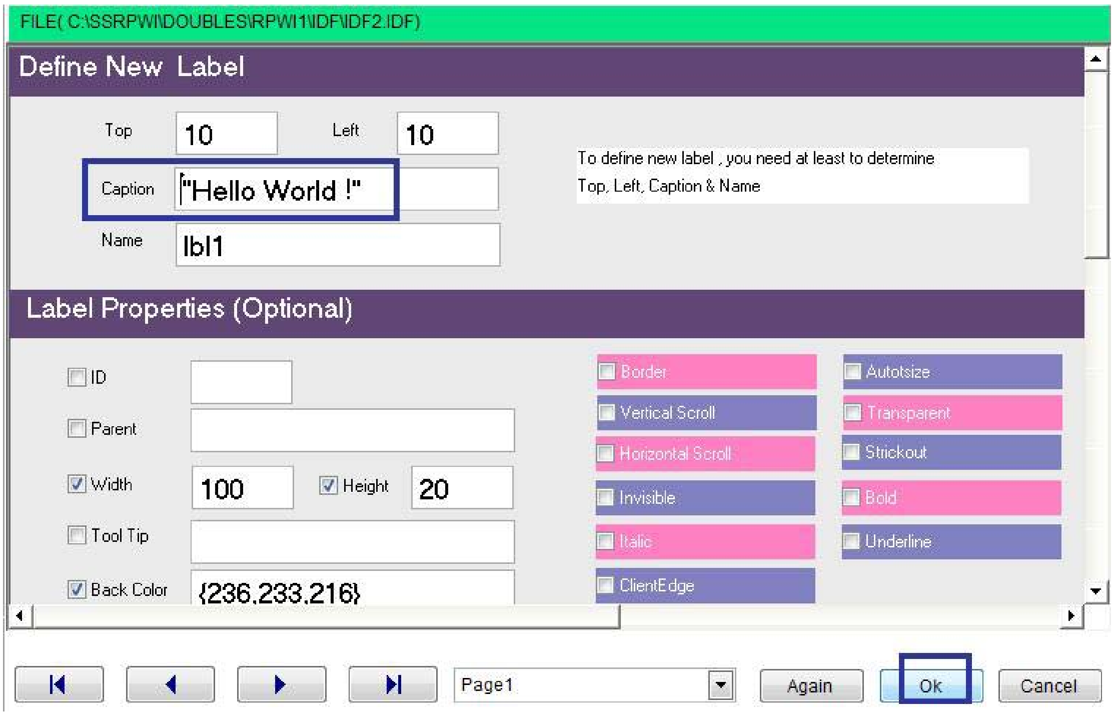
# Adding Control




Steps Tree



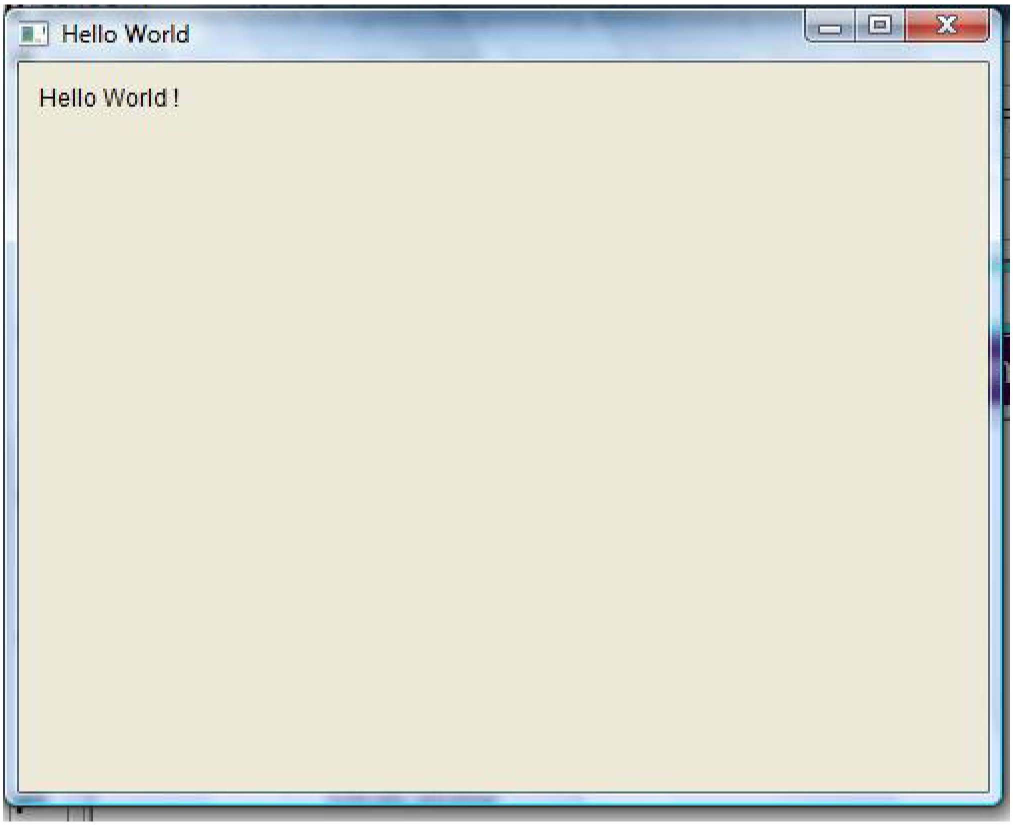

Domain (Controls) Component (Label)




Interaction Page

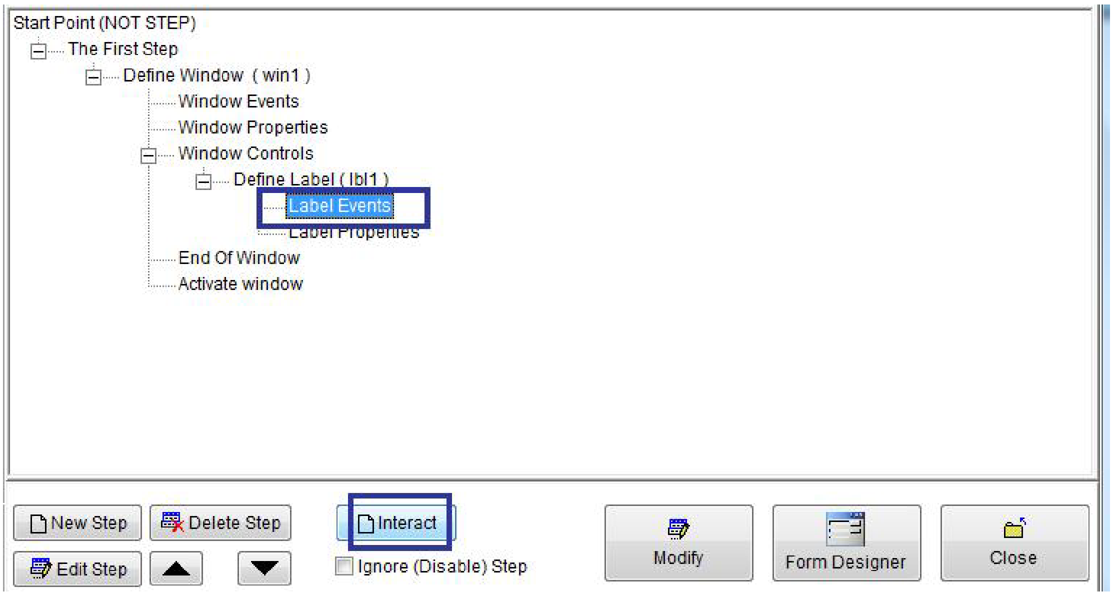



Final Steps Tree

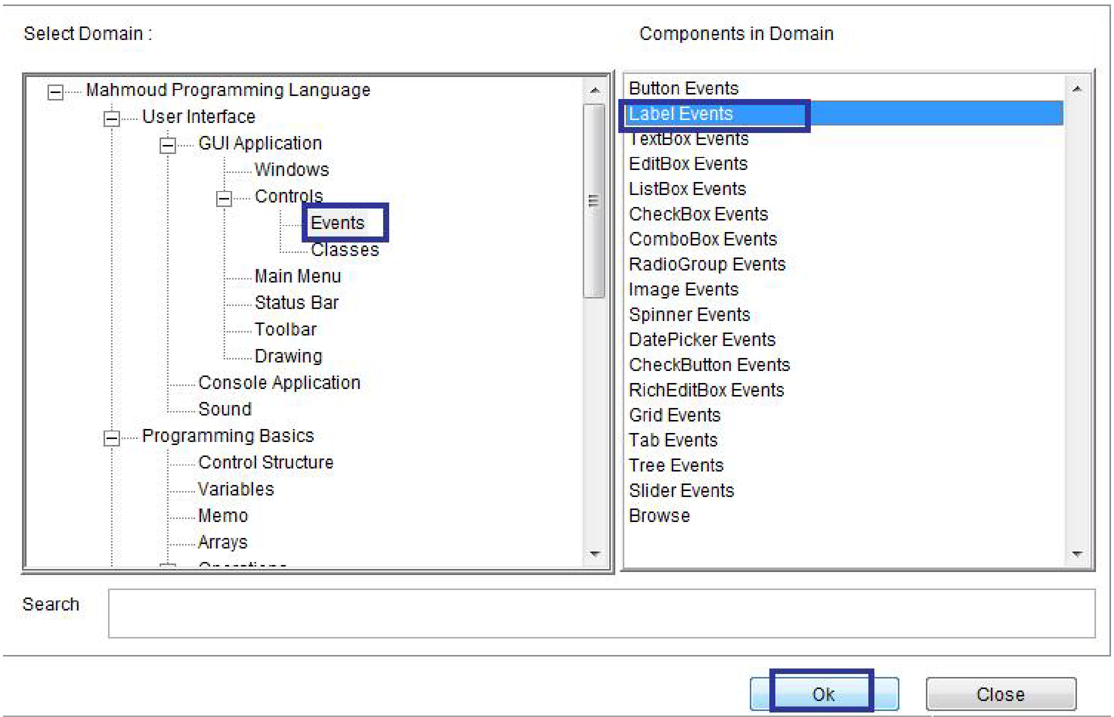


The Final Application

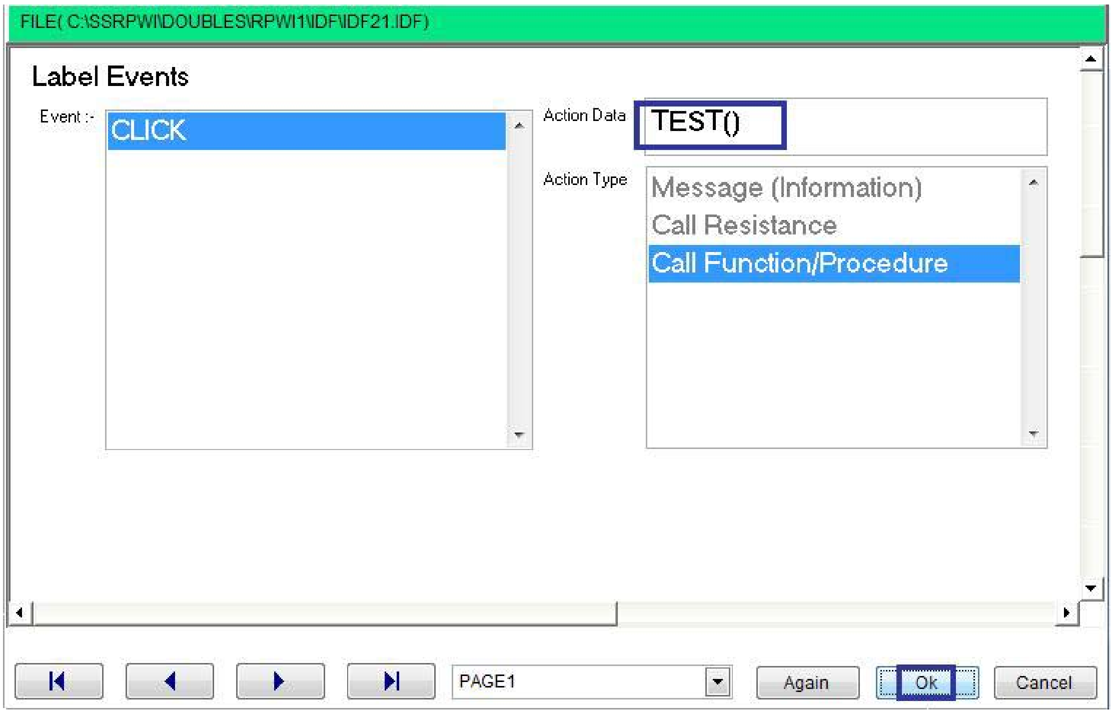
## Control Events & Control Class



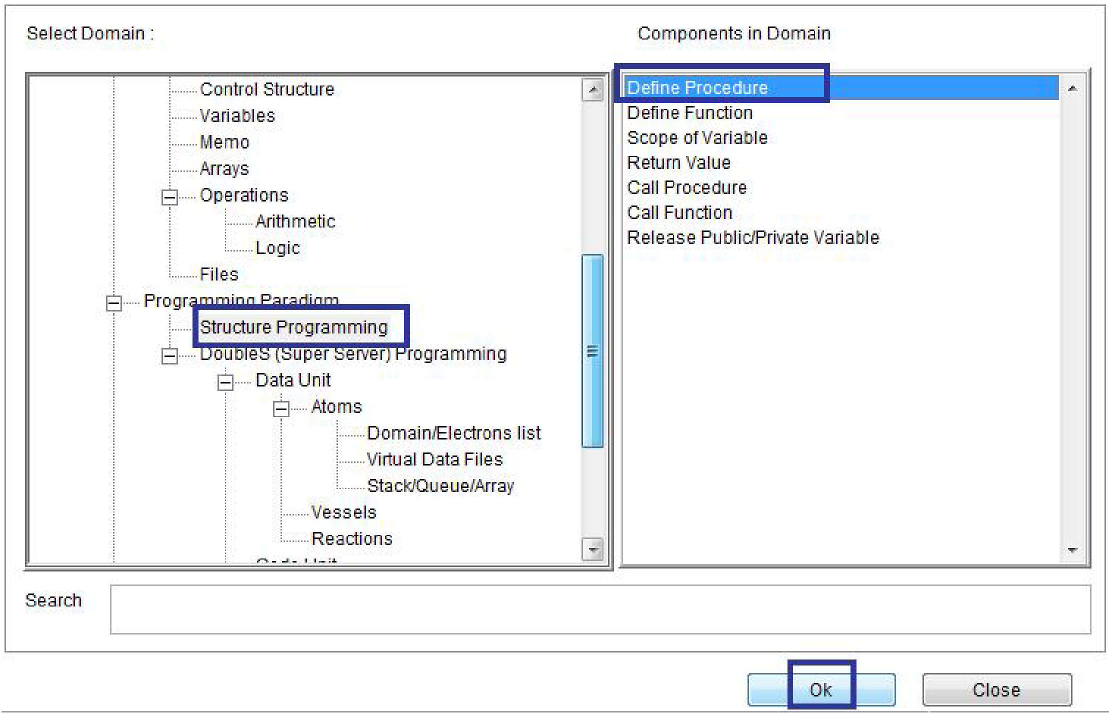
Steps Tree



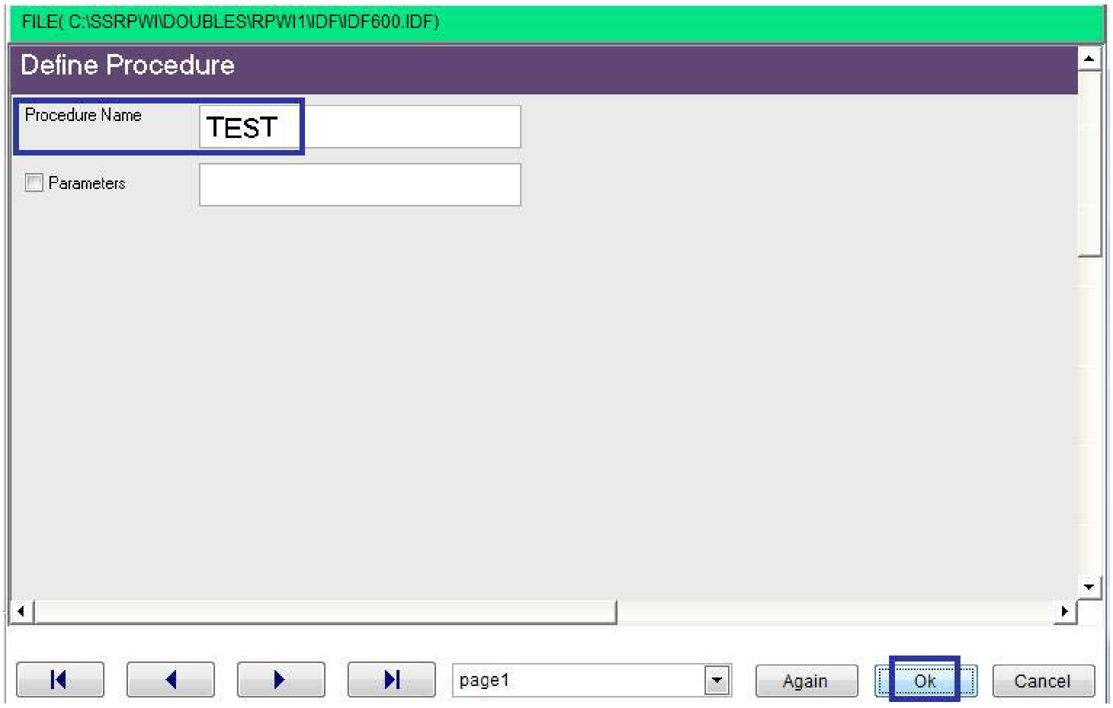

Domain(Events) Component (Label Events)



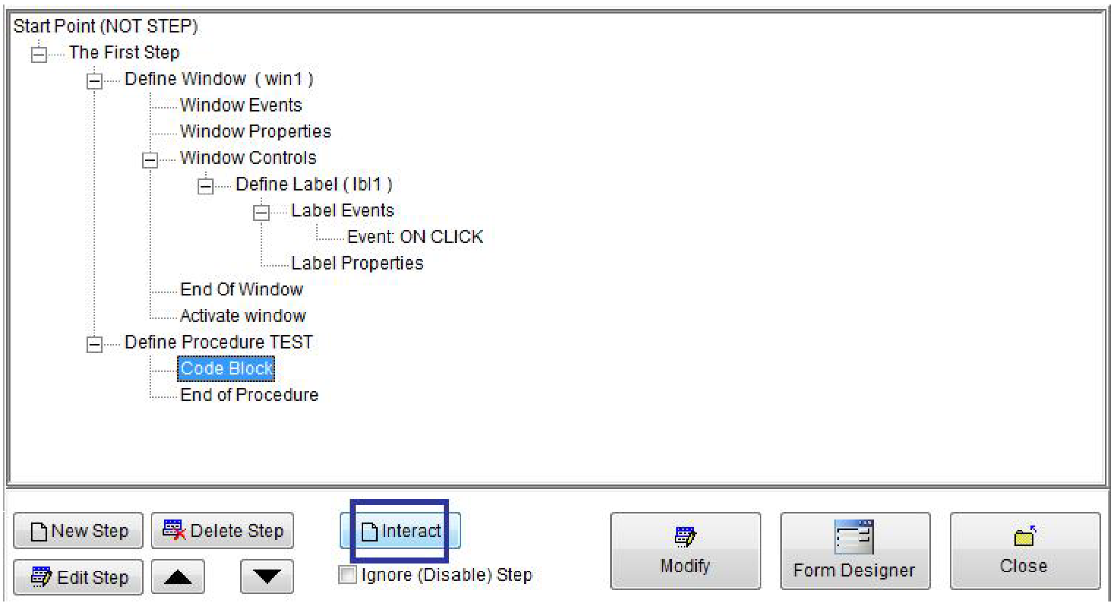

Interaction Page



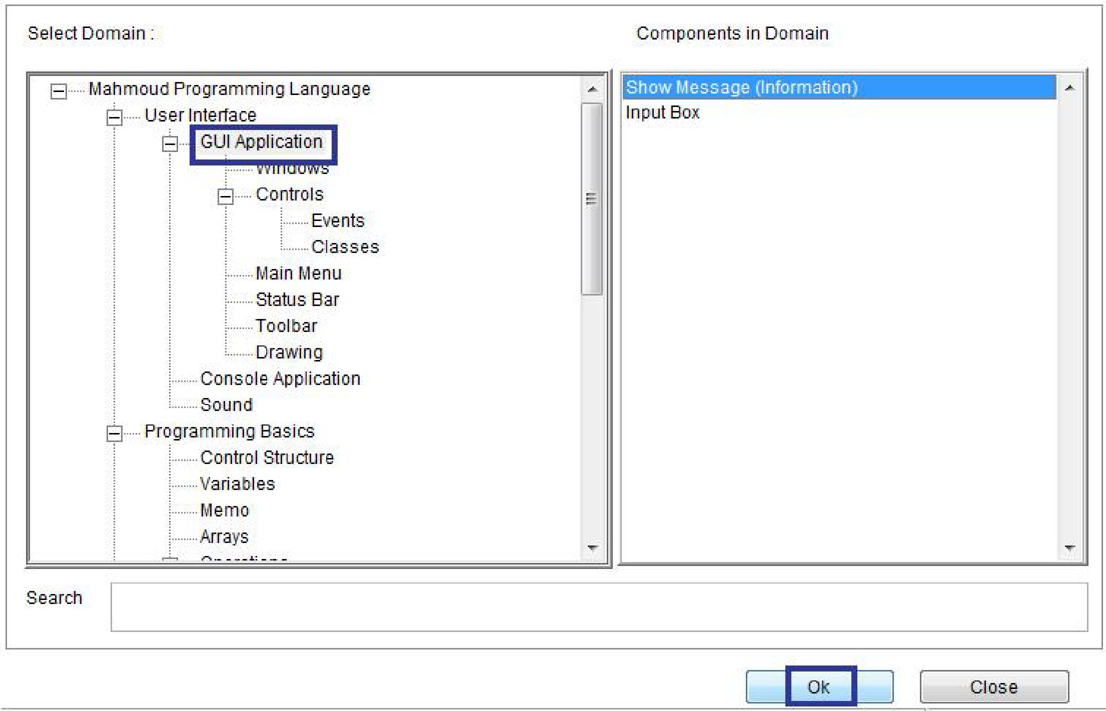

Domain (Structure Programming) Component (Define Procedure)



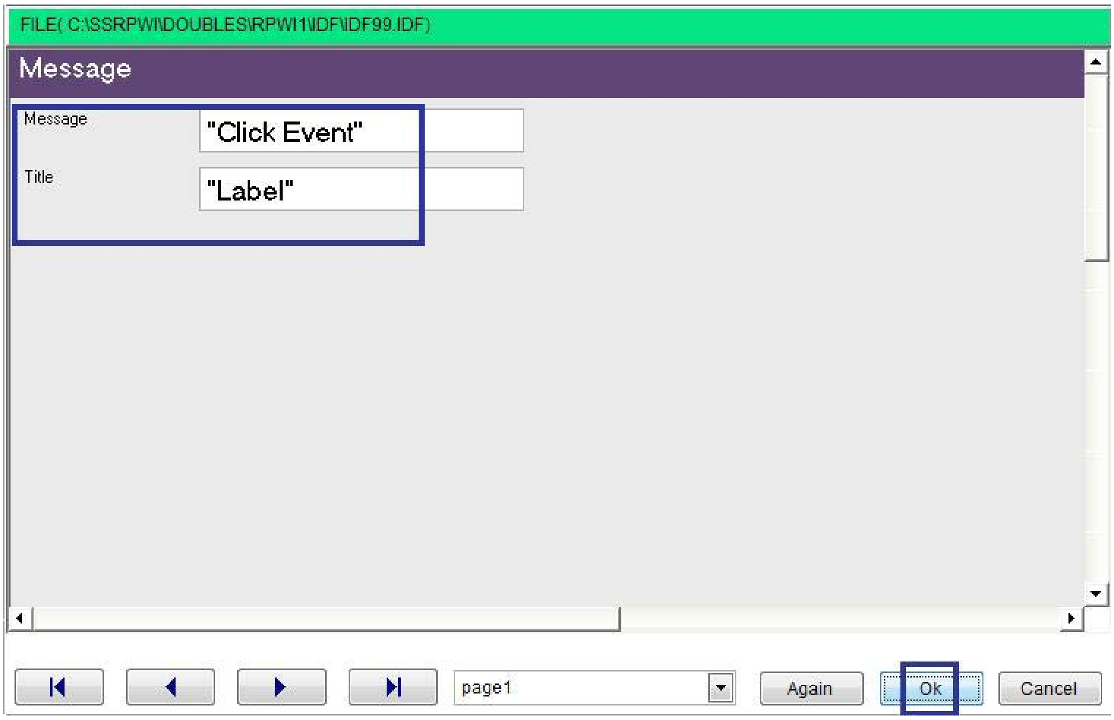

Interaction Page



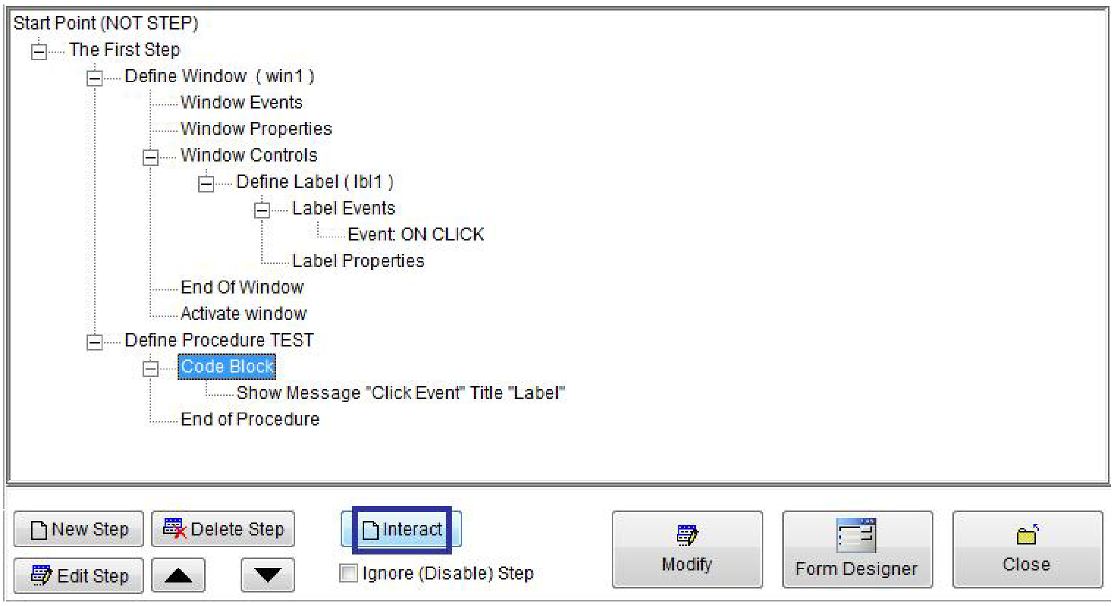

Steps Tree



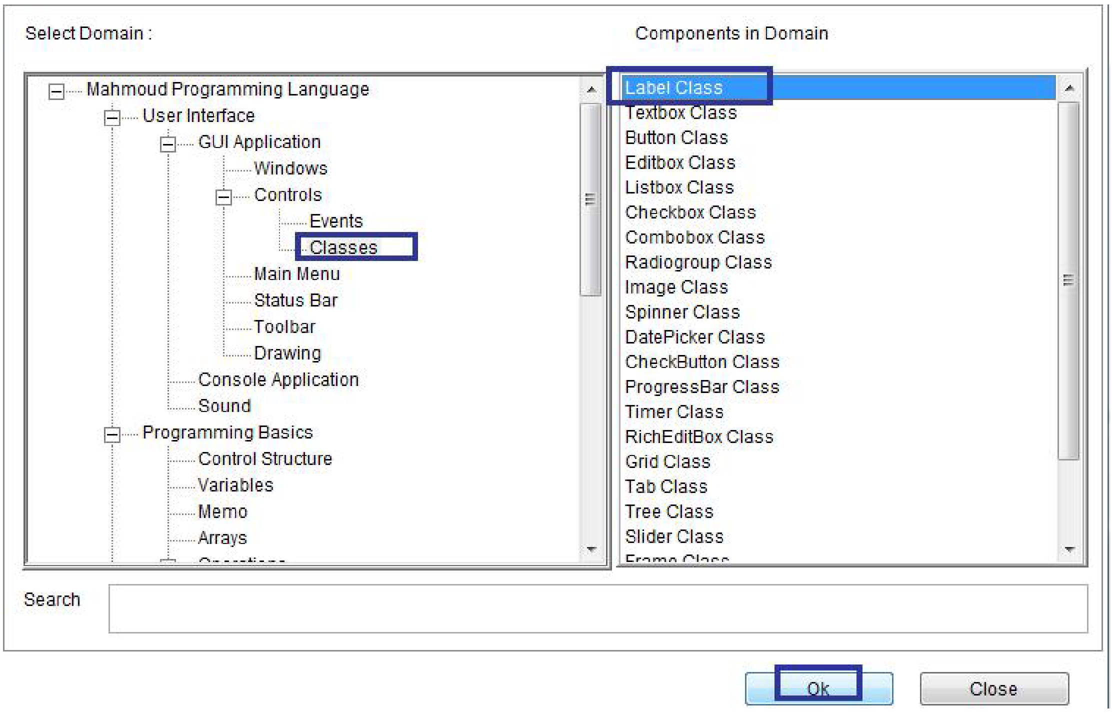

Domain (GUI Application) Component (Show Message)



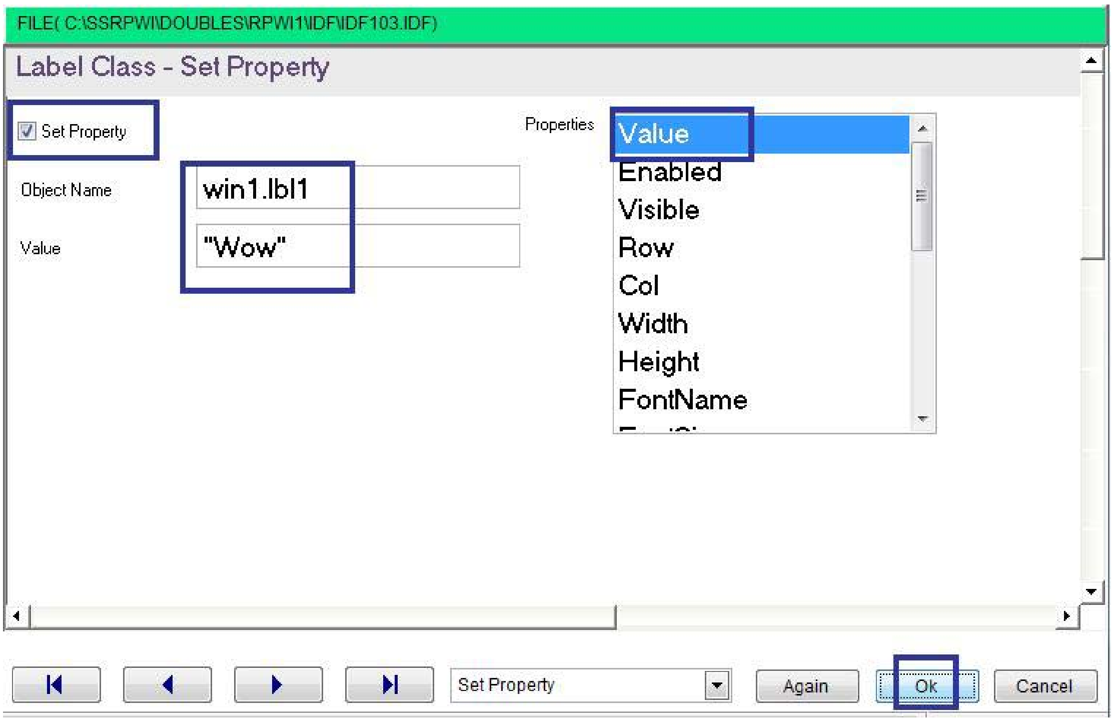
Interaction Page



Steps Tree



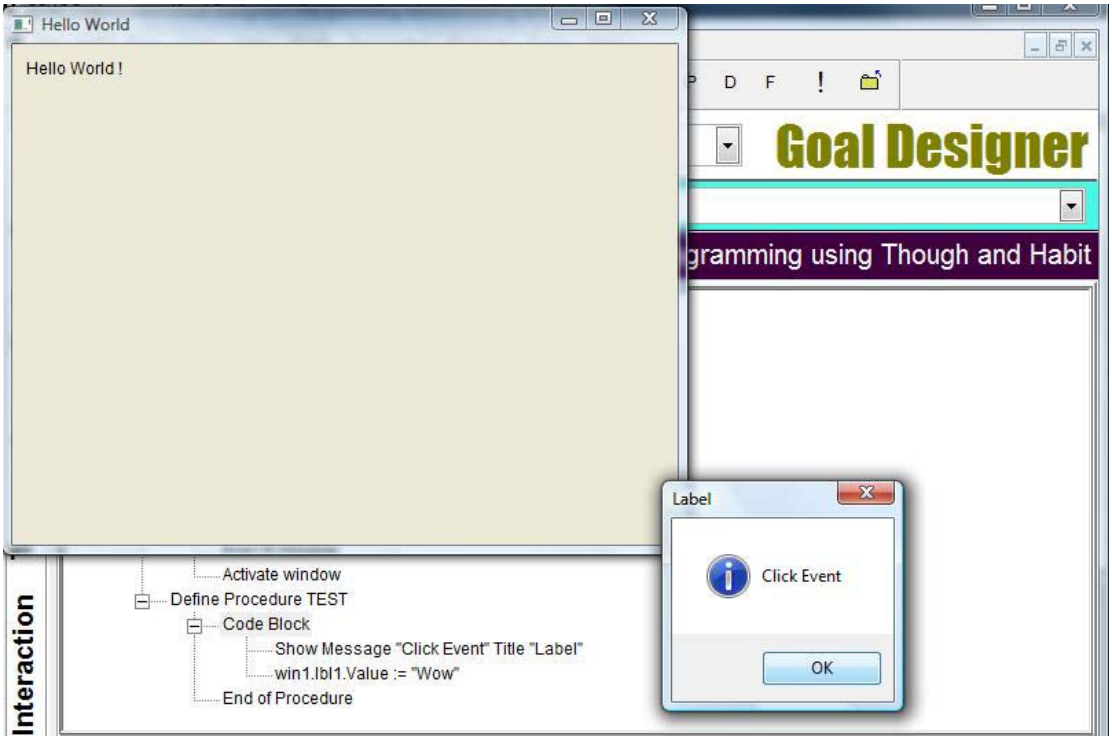
Domain (Classes) Component (Label Class)



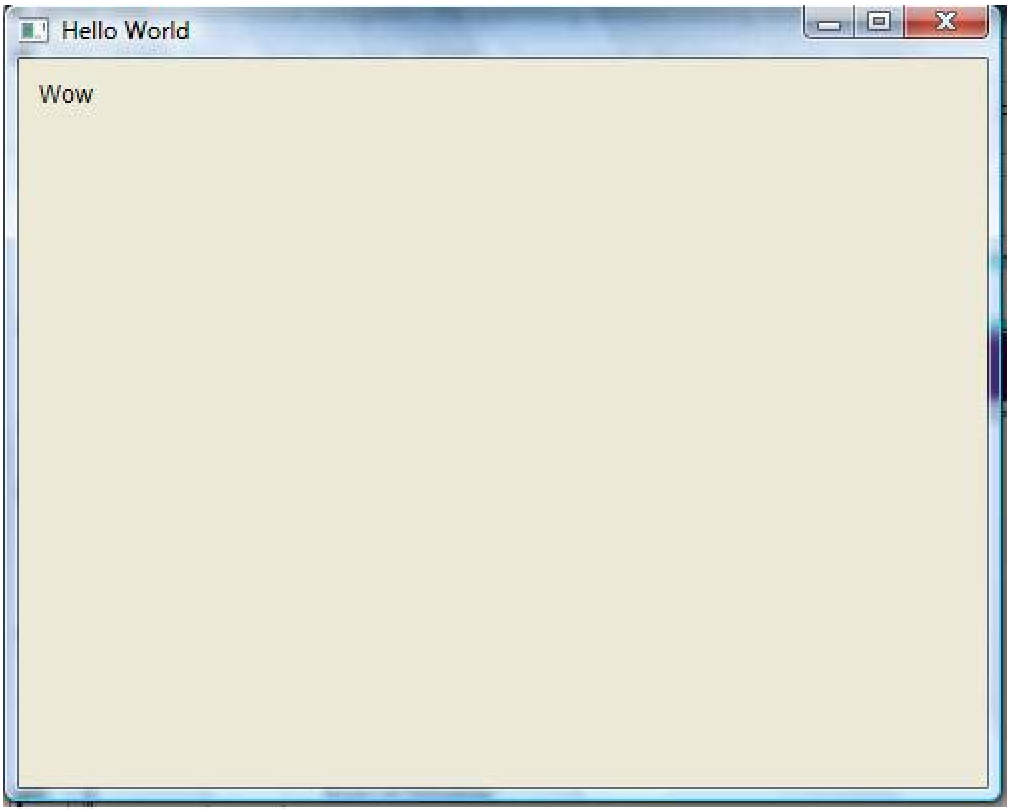
Interaction Page




Steps Tree

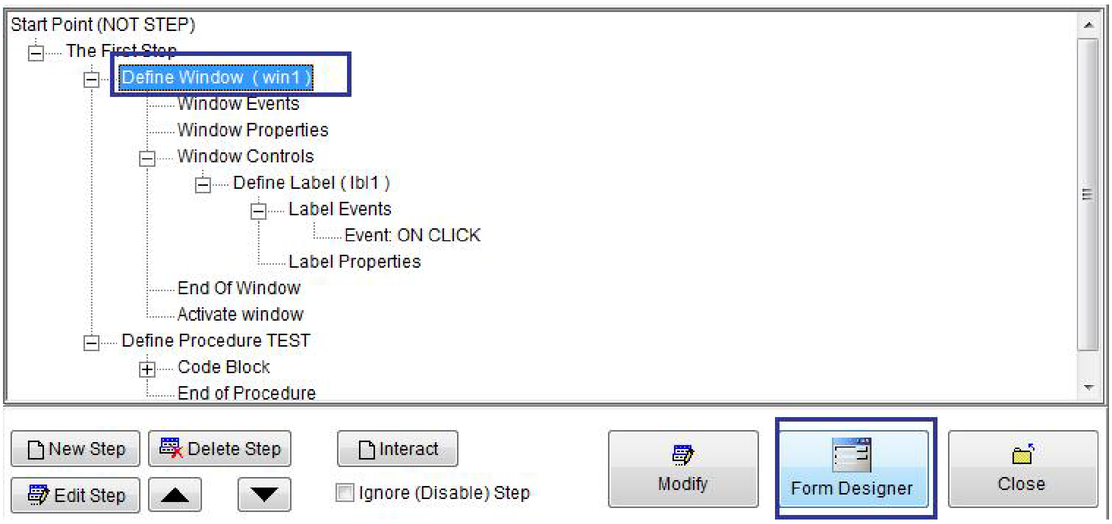



The Final Application

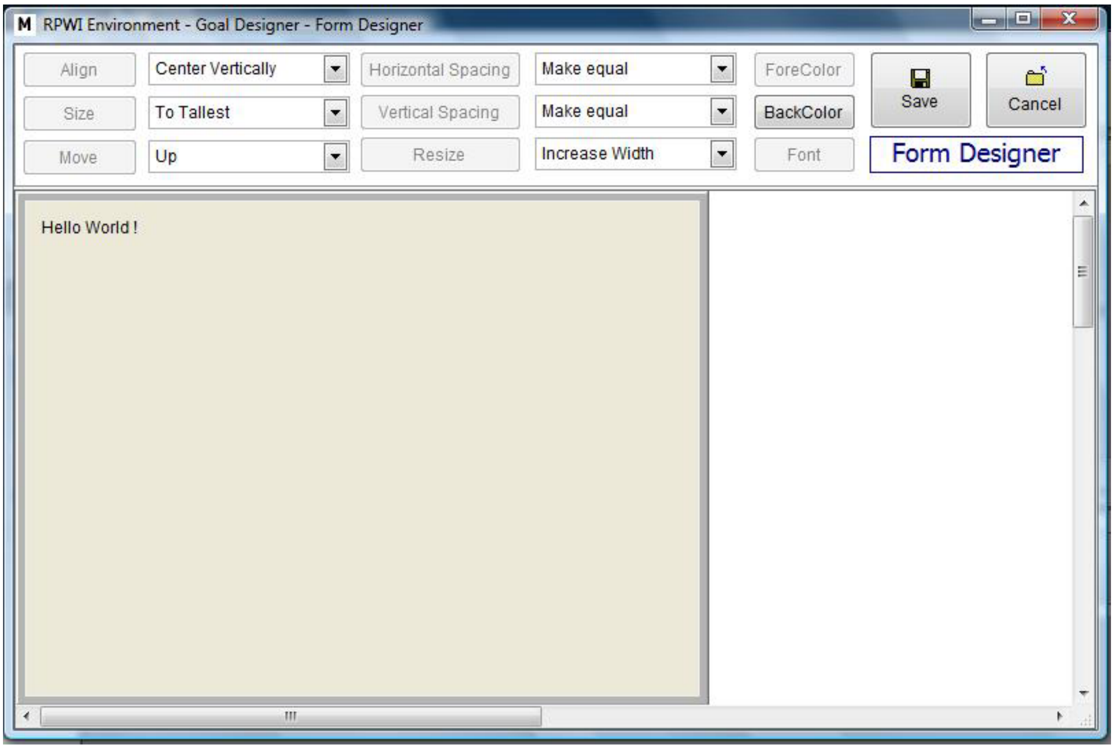


The Final Application

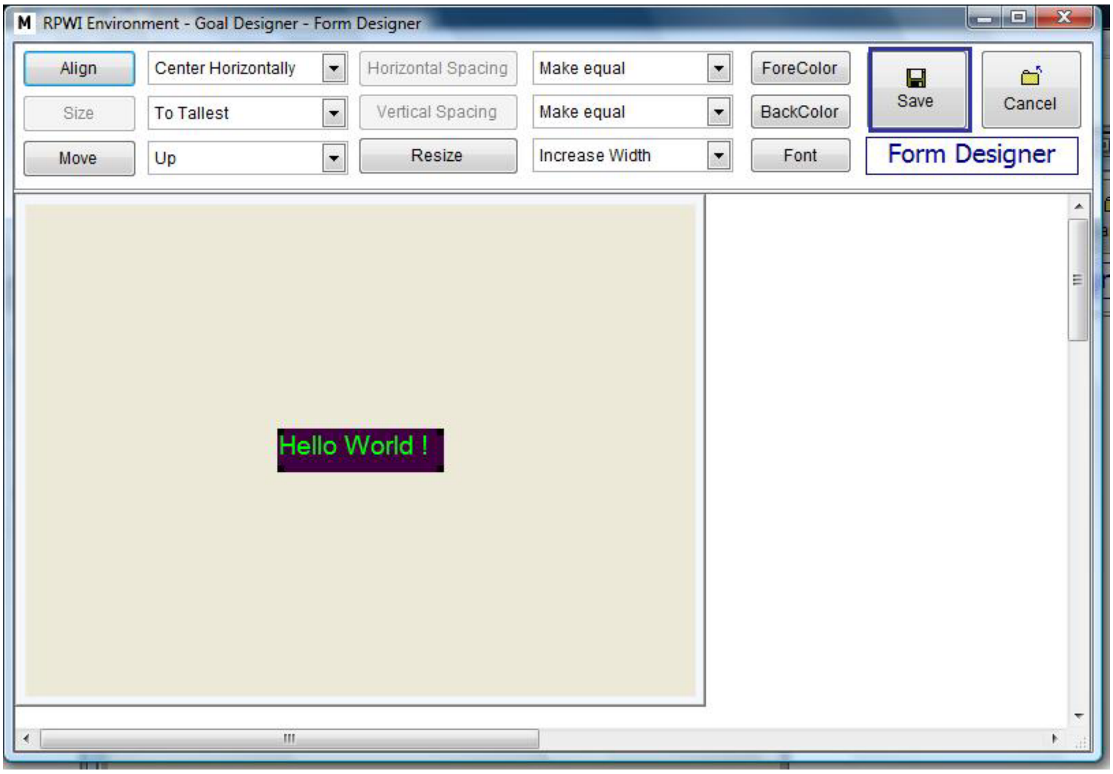
# Form Designer



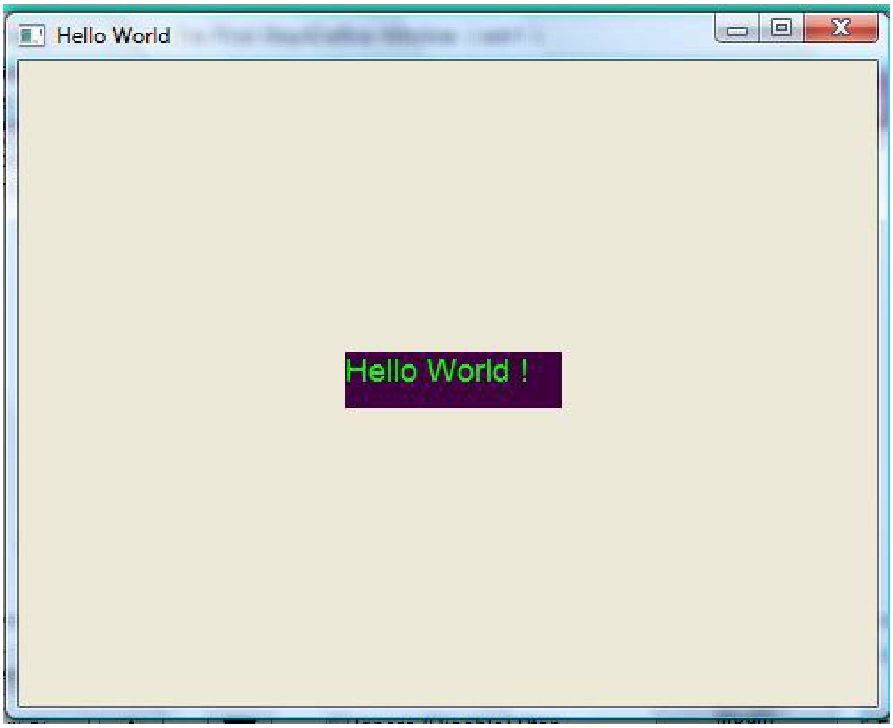
Steps Tree




Form Designer




Form Designer

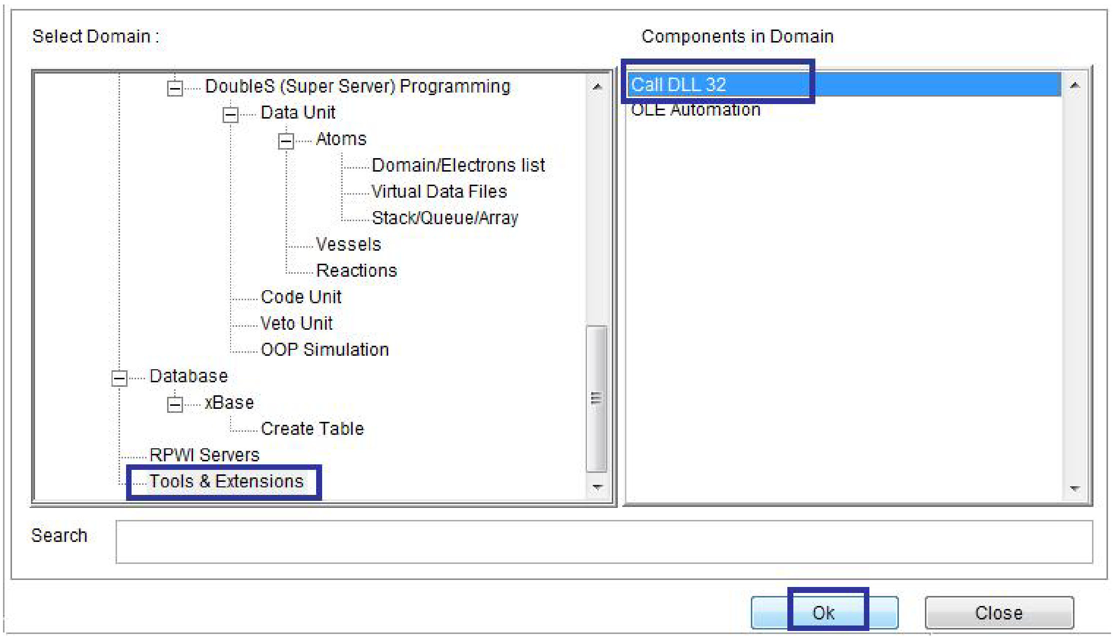



The Final Application

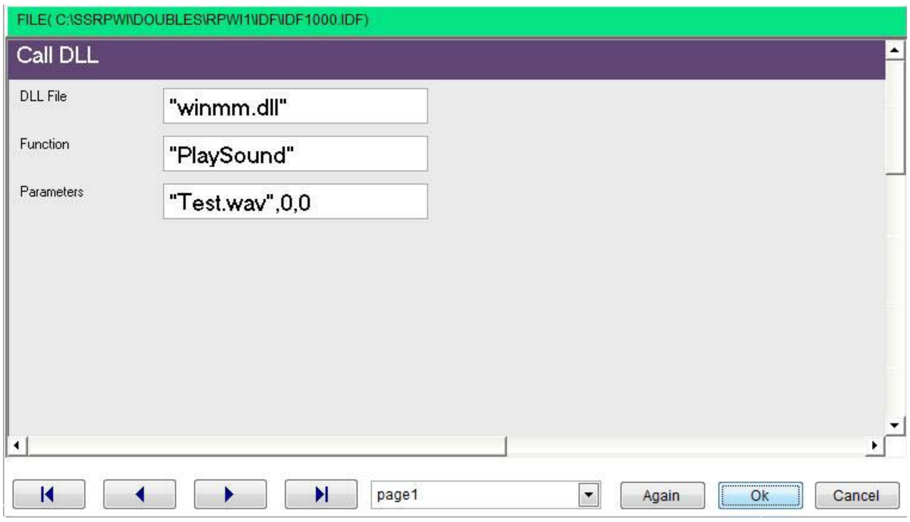
# Language Extension

- Calling DLL Functions
- OLE Automation

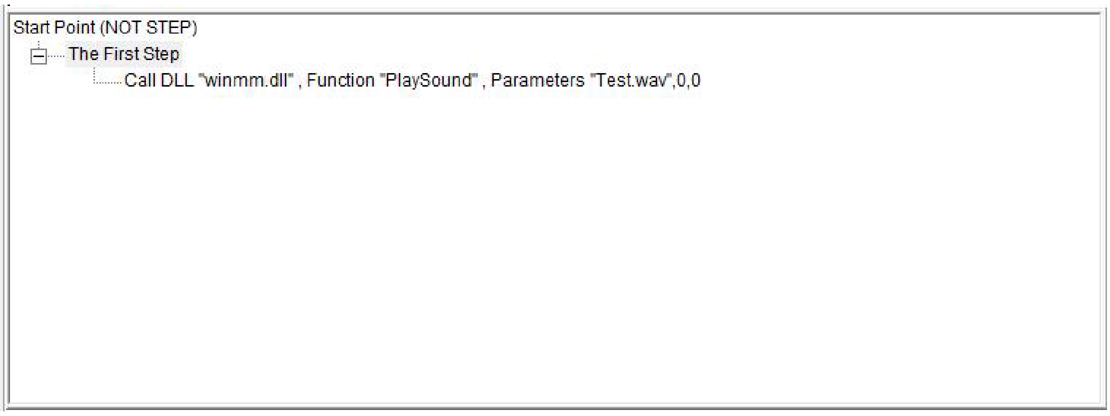
## Calling DLL Functions




Domain (Tools & Extensions) Component (CALL DLL 32)

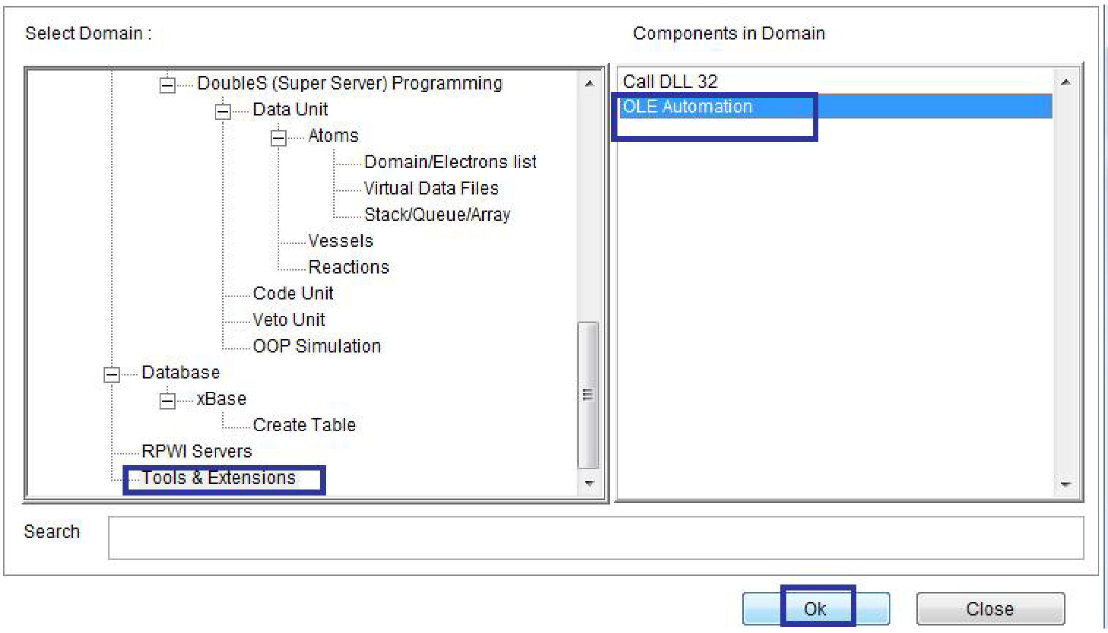



Interaction Page

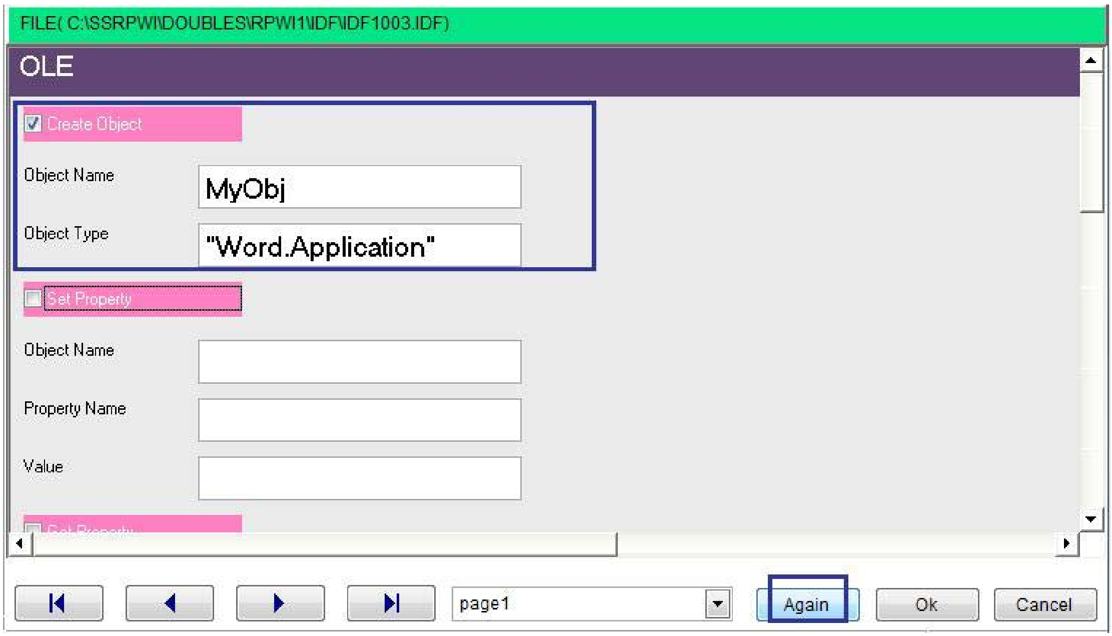



Steps Tree

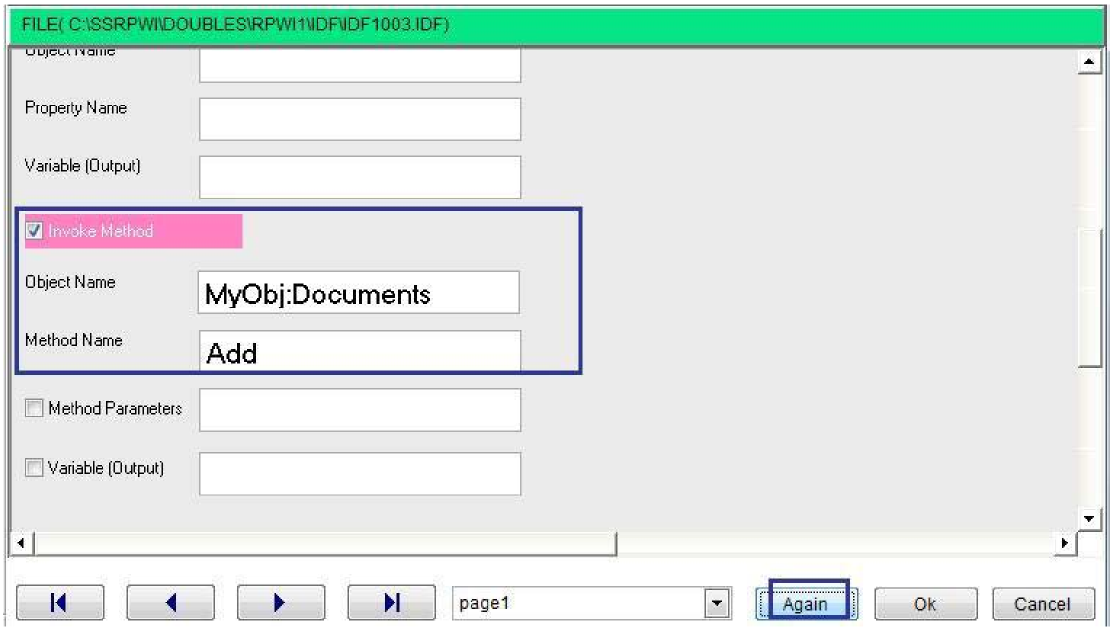
# OLE Automation



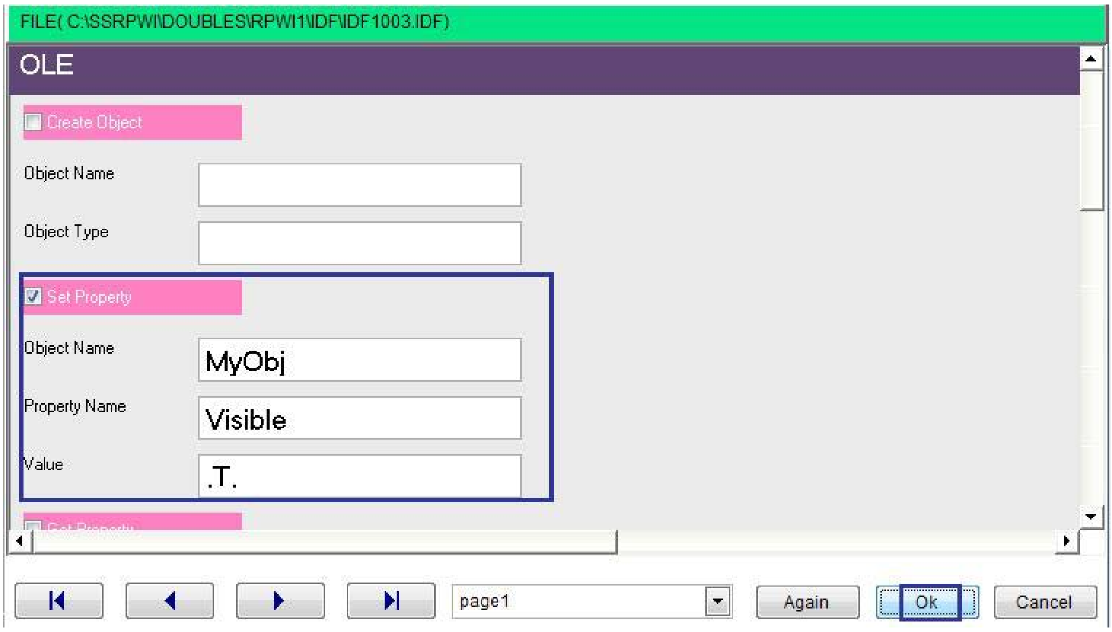

Domain (Tools & Extensions) Component (OLE Automation)




Interaction Page



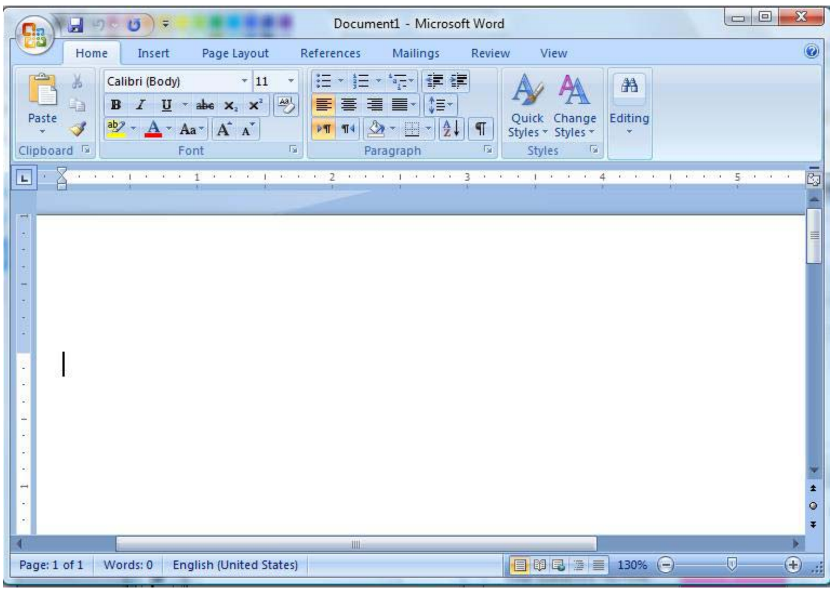

Interaction Page




Interaction Page




Final Steps Tree




Microsoft Word